VMware Cloud Director API Programming Guide for Service Providers

VMware Cloud Director API 38.0

18 JUL 2023

VMware Cloud Director 10.5

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

VMware Cloud Director™ API Programming Guide for Service Providers

This edition of the *VMware Cloud Director API Programming Guide* provides information about version 38.x of the VMware Cloud Director API.

VMware provides many different APIs and SDKs for applications and goals. This guide provides information about the VMware Cloud Director API for developers who are interested in creating RESTful clients of VMware Cloud Director.

Changes to API Login and Logout

Starting with VMware Cloud Director API version 38.0, the following APIs are no longer available. You must log in through the corresponding `cloudapi` login. VMware Cloud Director API version 37.2 and earlier continue to support these APIs as per the backward compatibility commitment of VMware Cloud Director.

Operation	Description
`POST /api/sessions`	Create a login session.
`DELETE /api/session`	Delete the current user session. **Note** `GET /api/session` is not deprecated and VMware Cloud Director API version 38.0 continues to support it.

Revision History

The VMware Cloud Director API version number is incremented whenever any of its types or operations changes. The *VMware Cloud Director API Programming Guide* is revised with each release of VMware Cloud Director. Versions of the VMware Cloud Director API that were not introduced in a VMware Cloud Director release are documented in the *vCloud Air Compute Service Programming Guide* . An API version can be deprecated whenever a newer version of the API supports an equivalent set of features. Use of a deprecated API version is not recommended. Support for an API version may be removed in the release following deprecation.

The life cycle of minor VMware Cloud Director API versions depends on the major product API version. For example, VMware Cloud Director API Version 35.2 reaches end of general support at the same time as VMware Cloud Director API Version 35.0. See the VMware Cloud Director API version life cycle table on https://docs.vmware.com/en/VMware-Cloud-Director/index.html.

Intended Audience

This guide is intended for software developers who are building VMware Ready Cloud Services, including interactive clients of VMware Cloud Director. This guide discusses Representational State Transfer (REST) and RESTful programming conventions, the Open Virtualization Format Specification, and VMware Virtual machine technology. You must be familiar with these and other widely deployed technologies such as XML, HTTP, and the Windows or Linux operating system.

Related Publications

The *VMware Cloud Director API Schema Reference* includes reference material for all elements, types, queries, and operations in the VMware Cloud Director API. It also includes the schema definition files. The schema reference is available at https://developer.vmware.com/.

The *VMware Cloud Director Service Provider Admin Guide* and *VMware Cloud Director Tenant Guide* contain detailed information about many of the objects and operations referred to in this guide. Most users of the VMware Cloud Director API will find the information in those documents valuable when developing client applications. To access the current versions of these and other VMware publications, go to https://docs.vmware.com/.

Starting with version 9.5, VMware Cloud Director integrates OpenAPI. For information about VMware Cloud Director OpenAPI, see *Getting Started with VMware Cloud Director OpenAPI* at https://developer.vmware.com/.

About the VMware Cloud Director API

1

The VMware Cloud Director API provides support for developers who are building interactive clients of VMware Cloud Director™ using a RESTful application development style.

VMware Cloud Director API clients communicate with servers over HTTP, exchanging representations of VMware Cloud Director objects. These representations take the form of XML elements. You use HTTP GET requests to retrieve the current representation of an object, HTTP POST and PUT requests to create or modify an object, and HTTP DELETE requests to delete an object.

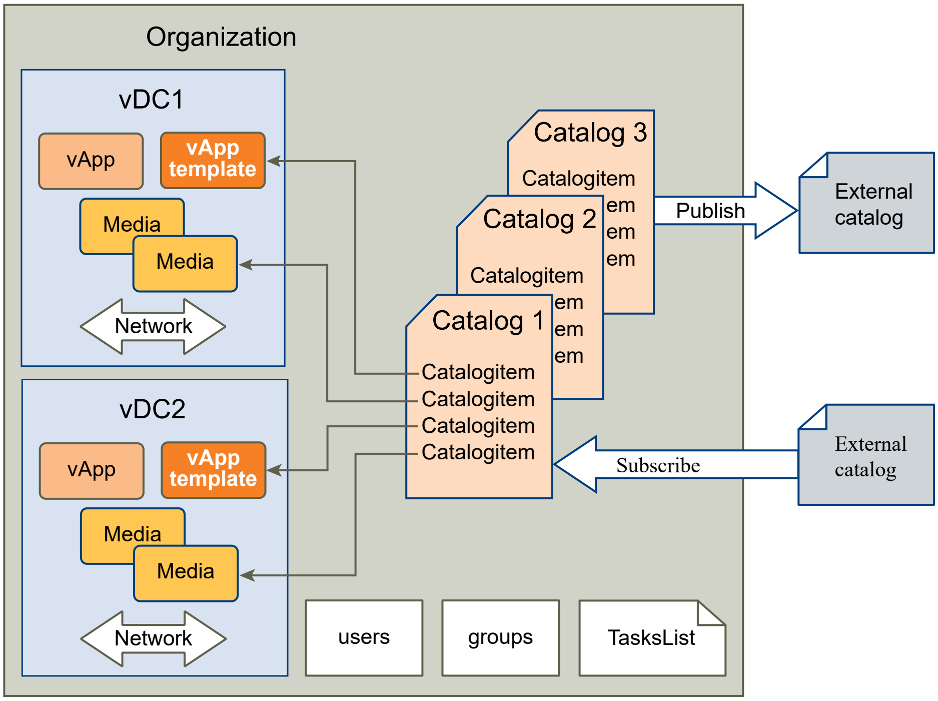
Read the following topics next:

- Object Taxonomy
- Objects, References, and Representations
- Links and Link Relations
- Client Workflow Overview
- Using the VMware Cloud Director API with VMware Cloud Director
- About the VMware Cloud Director API Examples

Object Taxonomy

The VMware Cloud Director API defines a set of objects common to cloud computing environments. An understanding of these objects, their properties, and their relationships is essential to using the VMware Cloud Director API.

Figure 1-1. VMware Cloud Director API Object Taxonomy



VMware Cloud Director API objects have the following high-level properties:

Organizations

A cloud can contain one or more organizations. Each organization is a unit of administration for a collection of users, groups, and computing resources. Users authenticate at the organization level, supplying credentials established when the user was created or imported. User credentials are authenticated by the organization's identity provider. VMware Cloud Director includes an integrated identity provider. It also supports several standards-based external identity providers.

Users and Groups

An organization can contain an arbitrary number of users and groups. Users can be created locally or managed by an external identity provider. Groups must be managed by an external identity provider. Permissions within an organization are controlled through the assignment of rights and roles to users and groups.

Catalogs

Catalogs contain references to vApp templates and media images. You can configure a catalog in several different ways:

- as a repository for local content that can remain private to the catalog owner or can be shared with other users, groups, or organizations in your cloud
- as a source of published content, to which other clouds can subscribe.

- as a local repository for content published by another cloud or any Web site that hosts a VMware Content Subscription Protocol (VCSP) endpoint.

An organization administrator or catalog owner controls catalog sharing. Organization administrators in organizations that have permission to publish catalogs control publication and subscription options for catalogs in their organization. A system administrator can enable background synchronization of catalogs with external sources and set background synchronization schedules to regulate consumption of network bandwidth by this activity.

Organization VDCs

An organization virtual datacenter (organization VDC) is a deployment environment for virtual systems owned by the containing organization, and an allocation mechanism for resources such as networks, storage, CPU, and memory. In an organization VDC, computing resources are fully virtualized, and can be allocated based on demand, service level requirements, or a combination of the two.

Organization VDC Networks

An organization VDC can be provisioned with zero or more networks. These organization VDC networks can be configured to provide direct or routed connections to external networks, or can be isolated from external networks and other organization VDC networks. Routed connections require an Edge Gateway and network pool in the VDC. The Edge Gateway provides firewall, network address translation, static routing, VPN, and load balancing services.

Virtual Systems and Media Images

Virtual systems and ISO-format media images are stored in a catalog and represented as catalog item objects. Virtual systems are stored as templates, using an open standard format (OVF 1.0). These templates can be retrieved from catalogs and transformed into virtual systems, called vApps, through a process called instantiation, which binds a template's abstract resource requirements to resources available in a VDC. A vApp contains one or more individual virtual machines (`Vm` elements), along with parameters that define operational details, including:

- How the contained virtual machines are connected to each other and to external networks.
- The order in which individual virtual machines are powered on or off.
- End-user license agreement terms for each virtual machine.
- Deployment lease terms, typically inherited from the containing organization, that constrain the consumption of VDC resources by the vApp.
- Access control information specifying which users and groups can perform operations such as deploy, power on, modify, and suspend on the vApp and the virtual machines that it contains.

Tasks

Asynchronous operations are tracked by task objects. Running and recently completed tasks initiated by members of an organization are kept on the organization's tasks list.

Objects, References, and Representations

The VMware Cloud Director API represents objects as XML documents in which object properties appear as elements and attributes with typed values. The object hierarchy is defined by an XML schema.

XML representations of first-class VMware Cloud Director API objects, such as the objects in Figure 1-1. VMware Cloud Director API Object Taxonomy, include these attributes.

id

The object identifier, expressed in URN format. The value of the `id` attribute uniquely identifies the object, persists for the life of the object, and is never reused. The `id` attribute value is intended to provide a context-free identifier that can be used with the VMware Cloud Director API `entityResolver` (see Retrieve an Object as an Entity).

type

The object type, specified as a MIME content type.

href

An object reference, expressed in URL format. This reference includes the object identifier portion of the `id` attribute value, and supplies additional information, including the current location of the object when accessed in a specific view. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat each `href` as an opaque string. The rules that govern how the server constructs `href` strings might change in future releases.

Views

The VMware Cloud Director API defines several contexts, or views, in which you can access objects in a cloud. These views are expressed in the URL returned as the `href` of an object, and have the following forms, where *API-URL* is a URL of the form `https://vcloud.example.com/api` and *object-type* is a string indicating the type of the object.

user view

A URL of the form `*API-URL*/*object-type*/*id*` indicates that any user can access the object.

admin view

A URL of the form `*API-URL*/admin/*object-type*/*id*` indicates that organization administrators and system administrators can access the object. Organization administrators do not have rights to modify some objects in the admin view.

extension view

A URL of the form *API-URL*/admin/extension/*object-type*/*id* indicates that system administrators can access the object.

A given object retrieved in one view may have a different representation and media type from the same object retrieved in a different view. Not all objects are presented in every view.

Example: Object id, type, and href Attributes

These abbreviated request and response examples show the id, type, and href attributes in the user and admin views of an organization.

Request:

```
GET https://vcloud.example.com/api/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4
```

Response:

```
<Org
   ...
   id="urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   type="application/vnd.vmware.vcloud.org+xml"
   href="https://vcloud.example.com/api/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   ...>
   ...
</Org>
```

The id value is the same in both cases, but the type and href attributes have values specific to the view.

Request:

```
GET https://vcloud.example.com/api/admin/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4
```

Response:

```
<AdminOrg
   ...
   id="urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   type="application/vnd.vmware.admin.organization+xml"
   href="https://vcloud.example.com/api/admin/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   ...>
   ...
</AdminOrg>
```

The value of the id attribute is a permanent, unique object identifier. The value of the href attribute is an object locator that refers to a specific view of the object in its current location. Unlike the value of the id attribute, object location and view context can change during the life of an object. The example in Using the entityResolver URL shows how to retrieve this object as entity.

When a client application must keep a persistent reference to an object, the best practice is to keep a reference to the `id` and the `href` (URL) that was most recently used to access the object. When the application needs to access the object in the future, it should first try using the saved `href`. If that fails, use the `id` with the entity resolver to obtain a valid reference to the object, then replace the saved `href` with that valid reference.

Links and Link Relations

The VMware Cloud Director API makes extensive use of `Link` elements to provide references to objects and the actions that they support. These elements are the primary mechanism by which a server tells a client how to access and operate on an object.

The server creates `Link` elements in a response body. They are read-only at the client. If a request body includes a `Link` element, the server ignores it.

Attributes of a Link Element

In the XML representation of a VMware Cloud Director object, each `Link` element has the following form:

```
<Link rel="relationship"
   type="application/vnd.vmware.vcloud.object_type+xml"
   href="URL"
   name="string"/>
```

Attribute values in a `Link` element supply the following information:

rel

Defines the relationship of the link to the object that contains it. A relationship can be the name of an operation on the object, a reference to a contained or containing object, or a reference to an alternate representation of the object. The relationship value implies the HTTP verb to use when you use the link's `href` value as a request URL.

type

The object type, specified as a MIME content type, of the object that the link references. This attribute is present only for links to objects. It is not present for links to actions.

href

An object reference, expressed in URL format. This reference includes the object identifier portion of the `id` attribute value, and supplies additional information, including the current location of the object when accessed in a specific view. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat each `href` as an opaque string. The rules that govern how the server constructs `href` strings might change in future releases.

name

The name of the referenced object, taken from the value of that object's `name` attribute. Action links do not include a `name` attribute.

Table 1-1. Link Relationships and HTTP Request Types

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
abort	Abort this blocking task.	POST
add	Add an item to this container.	POST
add:dfwEnable	Enable distributed firewall	POST
alternate	References an alternate representation of this object.	GET
answer	Provide user input requested by a virtual machine.	POST
authorization:check	Check whether an extension service operation is authorized for an entity.	POST
blockingTask	A list of pending blocking task requests in this cloud.	GET
bundle:upload	Upload an extension service localization bundle.	PUT
bundles:cleanup	Remove unused extension service localization bundles.	POST
catalogItem	References the CatalogItem object that refers to this object.	GET
certificate:reset	Removes the SSL certificate used by this service.	POST
certificate:update	Updates the SSL certificate used by this service.	POST
checkCompliance	Check that this virtual machine is using a storage profile of the intended type.	POST
consolidate	Consolidate this virtual machine.	POST
controlAccess	Apply access controls to this object.	POST
copy	Reserved	N/A
customizeAtNextPowerOn	Force guest customization to be applied the next time this virtual machine is powered on.	POST
deploy	Deploy this vApp.	POST
disable	Disable this object.	POST
discardState	Discard the suspended state of this virtual machine.	POST
disk:attach	Attach an independent disk to this virtual machine.	POST
disk:detach	Detach an independent disk from this virtual machine.	POST
down	References an object contained by this object.	GET
down:aclRules	Retrieve the ACL rules for this resource class action.	GET

Table 1-1. Link Relationships and HTTP Request Types (continued)

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
down:apiDefinitions	Retrieve the API definitions for this extension service.	GET
down:apiFilters	Retrieve the API filters for this extension service.	GET
down:dfwSection	Retrieve the distributed firewall section of an Edge Gateway	GET
down:edge	Retrieve an Edge Gateway	GET
down:edgeOperations	Retrieve a list of Edge Gateway operations	GET
down:edges	Retrieve a list of Edge Gateways	GET
down:extensibility	Add an extension service to the system.	POST
down:fileDescriptors	Retrieve file descriptors for extension services APIs	GET
down:files	Retrieve files for extension services APIs	GET
down:resourceClassActions	Retrieve the actions defined for this extension service resource class.	GET
down:resourceClasses	Retrieve the resource classes defined by this extension service.	GET
down:serviceLinks	Retrieve the service links defined by this extension service.	GET
down:serviceResources	Retrieve the list of extension service resources of this class.	
down:services	Retrieve the list of registered extension services.	GET
download:alternate	Reserved	N/A
download:default	References the default location from which this file can be downloaded.	GET
download:identity	References the extended OVF descriptor of this vApp template. The extended OVF descriptor contains additional information such as MAC address, BIOS UUID, and `NetworkConfigSection`	GET
download:ovaDefault	References the default location from which this OVA file can be downloaded.	GET
download:ovaIdentity	References the OVA that contains the extended OVF descriptor of this vApp template. The extended OVF descriptor contains additional information such as MAC address, BIOS UUID, and `NetworkConfigSection`	GET
edgeGateway:configureServices	Update the network services offered by this Edge Gateway.	POST
edgeGateway:disableDistributedRouting	Disable the distributed logical routing feature on this Edge Gateway	POST

Table 1-1. Link Relationships and HTTP Request Types (continued)

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
edgeGateway:enableDistributedRouting	Enable the distributed logical routing feature on this Edge Gateway	POST
edgeGateway:reapplyServices	Reapply (after an update) the network services offered by this Edge Gateway.	POST
edgeGateway:redeploy	Redeploy the vShield Edge supporting this Edge Gateway.	POST
edgeGateway:syncSyslogSettings	Synchronize syslog server addresses used by this Edge Gateway with system defaults.	POST
edgeGateway:upgrade	Upgrade the backing configuration of this Edge Gateway from compact to full.	POST
edgeGateways	List the Edge Gateway objects in this organization VDC.	GET
edit	Modify this object, typically by replacing its current representation with the one in the request body.	PUT
enable	Enable this object.	POST
enterMaintenanceMode	Put this virtual machine into maintenance mode.	POST
entity	Retrieve a representation of the object on which an operation triggered this notification.	GET
entityResolver	Retrieve an object id as a context-free Entity element.	GET
event:create	Create an event in an this organization's event stream.	POST
exitMaintenanceMode	Take this virtual machine out of maintenance mode.	POST
fail	Fail this blocking task.	POST
firstPage	Reference to the first page of a paginated response.	GET
fanout:failed	Multisite authentication skipped for this request because the association status was ASYMMETRIC or UNREACHABLE. See Configuring and Managing Multisite Deployments.	GET
fanout:skipped	Multisite member status for this request was ACTIVE but authentication at the member failed for some other reason. See Configuring and Managing Multisite Deployments.	GET
installVmwareTools	Install VMware Tools on this virtual machine.	POST
instantiate	Instantiate a VDC template to create a VDC in this organization.	POST
keystore:reset	Removes the keystore used by this service.	POST
keystore:update	Updates the keystore used by this service.	POST
lastPage	Reference to the last page of a paginated response.	GET

Table 1-1. Link Relationships and HTTP Request Types (continued)

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
makeMandatory	Make a VM-host affinity rule mandatory	POST
makeOptional	Make a VM-host affinity rule optional	POST
media:ejectMedia	Eject virtual media from a virtual device.	POST
media:insertMedia	Insert virtual media into a virtual device.	POST
metrics	Retrieve a subset of current or historic metrics from a virtual machine	POST
merge	Merge one or more Provider VDCs with this Provider VDC.	POST
migrateVms	Migrate virtual machines from this resource pool to a different one.	POST
move	Reserved	N/A
nextPage	Reference to the next page of a paginated response.	GET
nsx	Entrypoint for VMware Cloud Director API for NSX	GET
orgVdcNetworks	List the organization VDC networks supported by this Edge Gateway.	GET
orgVdcNetwork:convertToInternalInterface	Convert this organization VDC network to an internal interface from a subinterface.	POST
orgVdcNetwork:convertToSubInterface	Convert this organization VDC network to an subinterface from an internal interface.	POST
ova	Reserved	N/A
ovf	References the OVF descriptor of this vApp template.	GET
power:powerOff	Power off this vApp or virtual machine.	POST
power:powerOn	Power on this vApp or virtual machine.	POST
power:reboot	Reboot this vApp or virtual machine.	POST
power:reset	Reset this vApp or virtual machine.	POST
power:shutdown	Shut down this vApp or virtual machine.	POST
power:suspend	Suspend this vApp or virtual machine.	POST
previousPage	Reference to the previous page of a paginated response.	GET
publish	Publish this extension to a specific tenant (Portal Extensibility)	POST
publish:all	Publish this extension to all tenants (Portal Extensibility)	POST
publishToExternalOrganizations	Publish this catalog externally	POST

Table 1-1. Link Relationships and HTTP Request Types (continued)

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
recompose	Recompose this vApp to add, remove, or reconfigure virtual machines.	POST
reconfigureVm	Update multiple sections of a virtual machine.	POST
reconnect	Reconnect this vCenter Server to the system.	POST
refreshStorageProfiles	Refresh the list of storage profiles that exist on the vCenter service backing this Provider VDC.	POST
refreshVirtualCenter	Refresh the representation of this vCenter server	POST
register	Register a VCenter Server with the system.	POST
relinkToTemplate	Link a role to a role template	POST
reloadFromVc	Reload certain properties of this virtual machine from the vCenter database.	POST
relocate	Relocate this virtual machine.	POST
remove	Remove this object.	DELETE
remove:dfwDisable	Disable distributed firewall	POST
remove:force	Force removal of this object.	DELETE
repair	Repair this host or network.	POST
resourcePoolVmList	List the virtual machines using this resource pool.	GET
resume	Resume this blocking task.	POST
rights	List the service-specific rights created by this extension service.	GET
rights:cleanup	Remove service-specific rights no longer used by any extension service.	POST
screen:acquireTicket	Retrieve a screen ticket for this virtual machine.	GET
screen:thumbnail	Retrieve a thumbnail view of the screen of this virtual machine.	GET
self	References this object.	GET
shadowVms	List shadow virtual machines associated with the virtual machines in this vApp template.	GET
snapshot:create	Create a snapshot of the virtual machines in this vApp.	POST
snapshot:removeAll	Remove all snapshots created for the virtual machines in this vApp.	POST
snapshot:revertToCurrent	Revert all virtual machines in this vApp to their current snapshot.	POST

Table 1-1. Link Relationships and HTTP Request Types (continued)

rel Attribute Value	Action or Relationship Description	Implied HTTP Verb
storageProfile	References the storage profile for this object.	GET
subscribeToExternalCatalog	Add an external subscription to this catalog.	POST
sync	Synchronize this catalog or catalog item with its external source.	POST
syncSyslogSettings	Synchronize syslog server addresses used by this vApp network with system defaults.	POST
takeOwnership	Take ownership of this user's vApps, media, and catalogs.	POST
task	Retrieve the blocking task that triggered this notification.	GET
task:cancel	Cancel this task.	POST
task:create	Create a task object.	POST
task:owner	Reference to the owner of a task.	GET
tenants	Tenant list for VRO workflows.	GET
truststore:reset	Remove the truststore used by this service.	POST
truststore:update	Update the truststore used by this service.	PUT
undeploy	Undeploy this vApp.	POST
unlinkFromTemplate	Unlink a role from a role template	POST
unlock	Unlock this user account.	POST
unpublish	Remove this extension from a specific tenant (Portal Extensibility)	POST
unpublish:all	Remove this extension from all tenants (Portal Extensibility)	POST
unregister	Unregister this vCenter Server.	POST
up	References an object that contains this object.	GET
update:resourcePools	Update the resource pools of this Provider VDC	POST
updateProgress	Request an update of this task's progress.	POST
upgrade	Upgrade this host.	POST
upload:alternate	Reserved	N/A
upload:default	References the default location to which this object can be uploaded.	PUT
vSphereWebClientUrl	A URL that you can use to view this object with the vSphere Web Client	GET

Client Workflow Overview

VMware Cloud Director API clients implement a RESTful workflow, making HTTP requests to the server and retrieving the information they need from the server's responses.

About RESTful Workflows

REST, an acronym for Representational State Transfer, describes an architectural style characteristic of programs that use the Hypertext Transfer Protocol (HTTP) to exchange serialized representations of objects between a client and a server. In the VMware Cloud Director API, these representations are XML documents.

In a RESTful workflow, representations of objects are passed back and forth between a client and a server with the explicit assumption that neither party need know anything about an object other than what is presented in a single request or response. The URLs at which these documents are available often persist beyond the lifetime of the request or response that includes them. The other content of the documents is nominally valid until the expiration date noted in the HTTP `Expires` header.

VMware Cloud Director REST API Workflows

Application programs written to a REST API use HTTP requests that are often executed by a script or other higher-level language to make remote procedure calls that create, retrieve, update, or delete objects that the API defines. In the VMware Cloud Director REST API, these objects are defined by a collection of XML schemas. The operations themselves are HTTP requests, and so are generic to all HTTP clients.

To write a RESTful client application, you must understand only the HTTP protocol and the semantics of XML, the transfer format that the VMware Cloud Director API uses. To use the VMware Cloud Director API effectively in such a client, you need to know only a few things:

- The set of objects that the API supports, and what they represent; for example, what is a VDC and how does it relate to an organization or catalog?
- How the API represents these objects; for example, what does the XML schema for an `Org` look like? What do the individual elements and attributes represent?
- How a client refers to an object on which it wants to operate; for example, where are the links to objects in a VDC? How does a client obtain and use them?

You can find that information in this Guide, and in the *VMware Cloud Director API Schema Reference*. See About the Schema Reference.

RESTful Workflow Patterns

All RESTful workflows follow a common pattern.

1 Make an HTTP request, typically GET, PUT, POST, or DELETE. The target of this request is either a well-known URL such as the VMware Cloud Director API `versions` URL, or a URL obtained from the response to a previous request. For example, a GET request to an organization URL returns links to catalog and VDC objects that the organization contains.

2 Examine the response, which always includes an HTTP response code and usually includes a body. In the VMware Cloud Director API, a response body is an XML document that can contain any of the following items.

- XML elements and attributes that represent object properties
- `Link` elements that implement operations on the object or its contents
- If the object is being created or modified, an embedded `Task` object that tracks the progress of the creation or modification

These operations can repeat, in this order, for as long as necessary.

VMware Cloud Director API REST Requests

To retrieve object representations, clients make HTTP requests to object references. The server supplies these references as `href` attribute values in responses to GET requests.

Every VMware Cloud Director installation has a well-known URL from which an unauthenticated user can retrieve a `SupportedVersions` document, which lists each version of the VMware Cloud Director API that the server supports. For each version, the response lists the names and MIME types of the complex types defined in the version's XML namespace, and the version login URL. A system administrator can use that URL to authenticate to the cloud by logging in to the `System` organization. An authenticated user can discover other VMware Cloud Director API URLs by making GET requests to URLs retrieved from the login response, and the URLs contained in responses to those requests. See Chapter 3 Exploring a Cloud .

Requests are typically categorized by the type of requested operation: create, retrieve, update, and delete. This sequence of verbs is often abbreviated with the acronym CRUD. Each type of request is characterized by the use of specific HTTP verb to access a URL found in a `Link` element that has an operation-specific value for its `rel` (relation) attribute.

Table 1-2. CRUD Operations Summary

Operation Type	HTTP Verb	Link Relation	Operation Summary
Create	POST	add	Creates an object.
Retrieve	GET	down	Retrieves the representation of an existing object in its current state.

Table 1-2. CRUD Operations Summary (continued)

Operation Type	HTTP Verb	Link Relation	Operation Summary
Update	PUT	edit	Modifies an existing object.
Delete	DELETE	remove	Deletes an existing object. If the object is a container, you must remove all of its contents before you can delete it.

For example, this `Link` element indicates that you can use the URL https://vcloud.example.com/api/admin/org/26 to update the `Org` object that contains it.

```
<Link
   rel="edit"
   type="application/vnd.vmware.admin.organization+xml"
   href="https://vcloud.example.com/api/admin/org/26" />
```

The implied HTTP verb is PUT.

Important Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

Security

HTTP communications between a VMware Cloud Director API client and server are secured with SSL. API clients must also complete a login request to receive an authorization token that must be included in all subsequent requests.

Request Headers

The following HTTP headers are typically included in VMware Cloud Director API requests:

`Accept`

All requests must include an HTTP `Accept` header that specifies the type of response that is expected. Two forms of this header are supported:

```
Accept: application/*
```

```
Accept: application/vnd.vmware.vcloud.type+xml
```

The second form constrains acceptable responses to a *type* defined the VMware Cloud Director API schema. `Accept` headers of the following form are also allowed in VMware Cloud

Director API requests, but they are not compliant with RFC 2616 and might not be accepted by future versions of the VMware Cloud Director API.

```
Accept: application/*+xml
```

In addition, two key-value pairs are supported in the `Accept` header:

Table 1-3. Accept Header Key-Value Pairs

Key	Value	Requirement	Description
`version`	Any supported VMware Cloud Director API version number	Required	Specifies the VMware Cloud Director API version that the client is using.
`multisite`	`global`	Optional	Specifies that the request fans out to all member organizations. See Configuring and Managing Multisite Deployments.
	`local`	Optional	Specifies that the request does not fan out but includes multisite metadata included on fan-out. See Configuring and Managing Multisite Deployments.
	One or more location IDs	Optional	Multiple location IDs must be colon separated. Specifies that the request fans out only to the specified locations. See Configuring and Managing Multisite Deployments.

For example, the following header indicates that the request is from a VMware Cloud Director API version 30.0 client, and accepts any type defined in that API version:

```
Accept: application/*;version=30.0
```

In general, client requests can access objects defined by any version of the VMware Cloud Director API that is less than or equal to the version specified in the `Accept` header. See API Versions.

`Accept-Encoding`

By default, the system returns response content as uncompressed XML. Compressing the response can improve performance, especially when the response is large and network bandwidth is a factor. (Requests cannot be compressed.) To request a response to be returned as compressed XML, include the following header:

```
Accept-Encoding: gzip
```

The response is encoded using `gzip` encoding as described in RFC 1952, and includes the following header:

```
Content-Encoding: gzip
```

In the default configuration, responses smaller than 64 KB are never compressed.

Accept-Language

Message strings in `ErrorType` responses are localized. To specify the language desired in responses, use the `Accept-Language` request header. To request a response with message strings localized to French, use the following header:

```
Accept-Language: fr
```

Authorization

All requests to create a VMware Cloud Director API session must include an `Authorization` header of the form prescribed by the identity provider that your organization uses. See Create a VMware Cloud Director API Session. After you have established a session, you can use the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` and `X-VMWARE-VCLOUD-ACCESS-TOKEN` headers in the `Session` response to construct an `Authorization` header for use with subsequent requests.

Content-Type

Requests that include a body must include an appropriate HTTP `Content-Type` header. Content types for all elements are listed in the schema reference. In addition, the `type` attribute of a response body indicates the content type of the document. For example, this response fragment indicates that the content type associated with a `CatalogItem` object is `application/vnd.vmware.vcloud.catalogItem+xml`.

```
<CatalogItem
   type="application/vnd.vmware.vcloud.catalogItem+xml"
   name="Ubuntu Template with vsftpd"
... />
```

A POST or PUT request that supplies a `CatalogItem` in the request body requires the following `Content-Type` header:

```
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
```

When it appears as the value of a `Content-Type` header or the `type` attribute of an element in the VMware Cloud Director API, this string is case-insensitive in requests. The string can be returned in either mixed case or lowercase characters in responses.

`x-vcloud-authorization`

This header is returned with the `Session` response after a successful log-in to the integrated identity provider. As of API version 30, it is deprecated in favor of the `X-VMWARE-VCLOUD-ACCESS-TOKEN` value returned when you create a `Session`. If backward compatibility is a concern, you can supply `x-vcloud-authorization` instead of an `Authorization` header in most requests that do not fan out to members of an organization or site association. See Configuring and Managing Multisite Deployments.

`X-VMWARE-VCLOUD-AUTH-CONTEXT`

Unless you are making a request to the same organization that authenticated you, you must include this header in the request. You must specify a value that is the name of the organization to which you are making the request and that organization must be associated with the organization that authenticated you. See Configuring and Managing Multisite Deployments.

`X-VMWARE-VCLOUD-CLIENT-REQUEST-ID`

The value of this header is used to build a request ID returned in the value of the X-VMWARE-VCLOUD-REQUEST-ID header (see Response Headers). The value of this header cannot contain more than 128 characters drawn from the set of letters, numbers, and the hyphen (-). Values with invalid characters are ignored. Values with more than 128 characters are truncated.

Request Bodies

VMware Cloud Director uses a validating XML parser that requires elements in a request body to agree with the schema in order and number. Request bodies are rejected as invalid unless they meet the following criteria:

- XML namespace attributes must be supplied for all namespaces represented by elements in the request. See XML Namespace Identifiers.
- If multiple namespaces are represented in the request, XML namespace attributes must include an identifying prefix, and that prefix must be used with all elements from that namespace.
- All required elements must appear in request bodies. All elements that appear in request bodies must appear in the order that the schema establishes, and with content that conforms to the type constraint that the schema specifies.

Request Limits

To guard against denial-of-service attacks, VMware Cloud Director imposes the following limits on VMware Cloud Director API requests:

- Requests cannot exceed 512 KB.
- Requests cannot contain more than 4096 XML elements.
- Requests cannot have a depth greater than 100.

VMware Cloud Director API REST Responses

All responses include an HTTP status code and, unless the status code is 204 (No Content), a Content-Type header. Response content depends on the request. Some responses include a document body, some include only a URL, and some are empty.

Response Content

Response content depends on the requested operation. The response to a GET request is typically the complete representation of an existing object. The response to a PUT or POST request always contains values for the `href`, `name`, and `id` attributes of the object being created or updated. It also contains at most one `Task` element that you can retrieve to track the progress of the operation. When the `Task` completes with a status of `success`, a GET request to the object's `href` returns all properties of the object. If the `Task` completion status is not success, the object is in an indeterminate state, and should be deleted.

HTTP Response Codes

A VMware Cloud Director API client can expect a subset of HTTP status codes in a response.

Table 1-4. HTTP Status Codes that the VMware Cloud Director API Returns

Status Code	Status Description
`200 OK`	The request is valid and was completed. The response includes a document body.
`201 Created`	The request is valid. The requested object was created and can be found at the URL specified in the Location header.
`202 Accepted`	The request is valid and a task was created to handle it. This response is usually accompanied by a `Task` element.
`204 No Content`	The request is valid and was completed. The response does not include a body.
`400 Bad Request`	The request body is malformed, incomplete, or otherwise invalid.
`401 Unauthorized`	Any of: ■ Login failed. ■ Authentication token has expired ■ The user 's session has expired.

Table 1-4. HTTP Status Codes that the VMware Cloud Director API Returns (continued)

Status Code	Status Description
`403 Forbidden`	Any of: - One or more objects specified in the request could not be found in the specified container. - The user is not authenticated or does not have adequate privileges to access one or more objects specified in the request.
`404 Not Found`	Usually indicates a malformed request URL or request body.
`405 Method Not Allowed`	The HTTP method specified in the request is not supported for this object.
`406 Not Acceptable`	The resource identified by the request is not capable of generating a response of the type specified in the request's `Accept` header.
`409 Conflict`	The object state is not compatible with the requested operation.
`415 Unsupported Media Type`	The resource identified by the request does not support a request of the specified `Content-Type` and HTTP method.
`500 Internal Server Error`	The request was received but could not be completed because of an internal error at the server.
`503 The cell is currently busy and cannot accept incoming requests. Please try again later.`	The server is currently unable to handle the request due to too many requests. When the server recovers, the event `The cell accepting new requests` is posted.
`504 Gateway Timeout`	The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the request URL.

Response Headers

The following HTTP headers can appear in responses to VMware Cloud Director API requests:

`x-vcloud-authorization`

This header is returned with the `Session` response after a successful log-in to the integrated identity provider. As of API version 30, it is deprecated in favor of the `X-VMWARE-VCLOUD-ACCESS-TOKEN` value returned when you create a `Session`. If backward compatibility is a concern, you can supply `x-vcloud-authorization` instead of an `Authorization` header in most requests that do not fan out to members of an organization or site association. See Configuring and Managing Multisite Deployments.

`X-VMWARE-VCLOUD-REQUEST-ID`

The `X-VMWARE-VCLOUD-CLIENT-REQUEST-ID` header must contain only alpha-numeric characters or dashes. The header must contain at least one alpha-numeric character, and VMware Cloud Director shortens it if it's longer than 128 characters long. The `X-VMWARE-VCLOUD-REQUEST-ID` response header is formed from the first 128 characters of `X-VMWARE-VCLOUD-CLIENT-REQUEST-ID`, followed by a dash and a random UUID that the server generates. If the `X-VMWARE-VCLOUD-CLIENT-REQUEST-ID` header is invalid, null, or empty, the `X-VMWARE-VCLOUD-REQUEST-ID` is a random UUID. VMware Cloud Director adds this value to every VMware Cloud Director, vCenter Server, and ESXi log message related to processing the request, and provides a way to correlate the processing of a request across all participating systems. If a request does not supply a `X-VMWARE-VCLOUD-CLIENT-REQUEST-ID` header, the response contains an `X-VMWARE-VCLOUD-REQUEST-ID` header with a generated value that cannot be used for log correlation.

X-VMWARE-VCLOUD-ACCESS-TOKEN

The `Session` response to a successful login request includes an `X-VMWARE-VCLOUD-ACCESS-TOKEN` header whose value is an encoded key that you can use, along with the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` header, to construct an `Authorization` header to include in subsequent requests in place of the deprecated `x-vcloud-authorization` header.

X-VMWARE-VCLOUD-CEIP-ID

The presence of this header indicates that this site participates in VMware's Customer Experience Improvement Program ("CEIP"). Details regarding the data collected through CEIP and the purposes for which it is used by VMware are set forth in the Trust & Assurance Center at http://www.vmware.com/trustvmware/ceip.html. You can use the cell management tool to join or leave VMware's CEIP for this product at any time. See the "Cell Management Tool Reference" in the *VMware Cloud Director Administrator's Guide*.

X-VMWARE-VCLOUD-TOKEN-TYPE

If a response includes an `X-VMWARE-VCLOUD-ACCESS-TOKEN` header, it also includes an `X-VMWARE-VCLOUD-TOKEN-TYPE` header specifying the type of the `X-VMWARE-VCLOUD-ACCESS-TOKEN`.

X-VMWARE-VCLOUD-REQUEST-EXECUTION-TIME

The execution time, in milliseconds, of the request that generated this response.

Using the VMware Cloud Director API with VMware Cloud Director

VMware Cloud Director supports several versions of the VMware Cloud Director API. You can use a browser or other HTTP client program to send requests and receive responses.

The VMware Cloud Director REST API Reference documentation includes HTML reference material for all XML elements and complex types defined by the VMware Cloud Director API. It also includes example XML representations. See About the Schema Reference. For information about HTTP client programs to use with VMware Cloud Director, see REST Client Programs.

Procedure

1 Configure the VMware Cloud Director REST API base URL.

   a From the top navigation bar, select **Administration**.

   b In the left panel, under **Settings**, click **Public Addresses**.

   c Click **Edit** to customize the public endpoints.

   d To customize the VMware Cloud Director REST API and OpenAPI URLs, turn off the **Use Web Portal Settings** toggle.

   e Enter a custom HTTPS REST API base URL and click **Upload** to upload the certificates that establish the trust chain for that endpoint.

2 (Optional) Retrieve the list of supported API versions from the server.

   After the VMware Cloud Director REST API base URL has been configured, any HTTP client can request a document that lists all the API versions that the server supports. See Retrieve the Login URL and List of Supported API Versions.

3 (Optional) If you want to use the vSphere Client to access VMware Cloud Director API objects on a vSphere server, verify that the vSphere Client URL is enabled for all vCenter servers from which you want to retrieve the vSphere URL of an object.

Example: Retrieve the Login URL and List of Supported API Versions

The `api/versions` request can be made by any client, whether or not the client is authenticated by VMware Cloud Director. The response, a small subset of which is shown here, includes a `VersionInfo` element for each API version that this VMware Cloud Director installation supports. Each `VersionInfo` element contains:

- A `LoginUrl` element that contains the URL to which a client can make a login request to access that version of the VMware Cloud Director API. See Logging In.
- For API versions earlier than 5.7, `MediaTypeMapping` elements for each complex type supported by that version of the VMware Cloud Director API. This optional element is not returned in the `VersionInfo` for API version 5.7 and later.

Request:

```
GET http://vcloud.example.com/api/versions
```

Response:

```
200 OK
Content-Type: text/xml
...
<SupportedVersions
   xmlns="http://www.vmware.com/vcloud/versions"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.vmware.com/vcloud/versions
   http://vcloud.example.com/api/versions/schema/versions.xsd">
   <VersionInfo>
      <Version>5.1</Version>
      <LoginUrl>https://vcloud.example.com/api/login</LoginUrl>
      <MediaTypeMapping>
         <MediaType>application/vnd.vmware.vcloud.catalog+xml</MediaType>
         <ComplexTypeName>CatalogType</ComplexTypeName>
         <SchemaLocation>http://vcloud.example.com/api/v1.5/schema/master.xsd</SchemaLocation>
      </MediaTypeMapping>
      <MediaTypeMapping>
         ...
      </MediaTypeMapping>
      ...
   </VersionInfo>
   <VersionInfo>
      <Version>5.6</Version>
      <LoginUrl>https://vcloud.example.com/api/sessions</LoginUrl>
      <MediaTypeMapping>
         ...
      </MediaTypeMapping>
      ...
   </VersionInfo>
   <VersionInfo>
      <Version>5.7</Version>
      <LoginUrl>https://vcloud.example.com/api/sessions</LoginUrl>
   </VersionInfo>
</SupportedVersions>
```

Note You can use the URL in the `SchemaLocation` element with a GET request to retrieve the file in which that complex type is defined. This type of request can be made by any client, whether or not the client is authenticated by VMware Cloud Director. For example, this request retrieves the schema file `master.xsd`:

```
GET http://vcloud.example.com/api/v1.5/schema/master.xsd
```

All VMware Cloud Director API requests are processed in the `http://www.vmware.com/vcloud/v1.5` XML namespace. Schema files for all API versions can be retrieved from a `.../api/v1.5/...` URL.

What to do next

Decide on an HTTP client program to use. See REST Client Programs.

REST Client Programs

Any client application that can send HTTPS requests can be an appropriate tool for developing RESTful applications with the VMware Cloud Director API.

REST client plug-ins are available for most browsers and many IDEs. Many of the examples in the *VMware Cloud Director API Programming Guide* were developed using two open-source programs: cURL (http://curl.haxx.se/) and the RESTclient (http://code.google.com/p/rest-client/).

VMware provides additional SDK products that implement language-specific bindings for the VMware Cloud Director API, and include their own HTTP client capability. See http://communities.vmware.com/community/developer/forums.

About the Schema Reference

The *vCloud Air API Schema Reference* includes reference material for all elements, types, queries, and operations in the VMware Cloud Director API. It also includes a downloadable set of the schema definition files.

The *vCloud Air API Schema Reference* is available in HTML at https://code.vmware.com.

Important The schema reference includes reference topics for the entire VMware Cloud Director API, including topics that apply to objects and operations that are accessible only to vCloud Air tenants.

About the VMware Cloud Director API Examples

The *VMware Cloud Director API Programming Guide* includes many examples of HTTP requests and responses. These examples show the workflow and content associated with operations such as browsing, provisioning, and managing your cloud and its contents, and operating virtual systems.

Example requests generally conform to the rules listed in Request Bodies. Most example responses show only those elements and attributes that are relevant to the operation being discussed. Ellipses (…) indicate omitted content within response bodies. Several additional conventions apply.

- The HTTP `Accept` header, which is required in all requests, is omitted from most examples. See API Versions for more about this header and how it is used by the VMware Cloud Director API.
- Authorization headers such as `x-vcloud-authorization` are omitted from most examples. See VMware Cloud Director API REST Requests for more about how authorization headers are used by the VMware Cloud Director API.
- All other request headers required by the VMware Cloud Director API are included in example requests that are not fragments of some larger example. Although the examples show these strings using the character case in which the implementation defines them, header names and values are case-insensitive, and can be submitted or returned in any character case. Other HTTP headers, such as `Date`, `Content-Length`, and `Server`, are omitted because they are not relevant to the specifics of any example.
- The XML version and encoding header

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  ```

  is included in example requests but omitted from example responses.

- In most examples, object IDs shown in `href` attribute values appear as small integers, for example `vapp-7` or `org/3`. In the VMware Cloud Director API that VMware Cloud Director supports, object IDs are universal unique identifiers (UUIDs) as defined by RFC 4122, for example `vapp-f5e185a4-7c00-41f1-8b91-0e552d538101` or `org/89a1a8f9-c518-4f53-960c-950db9e3a1fd`. Examples that show `Role` or `Right` objects use the actual UUIDs for roles and rights, which are invariant across installations.

Required Roles and Rights

Where a topic includes an example, it specifies a prerequisite role that normally has the rights required to run the example. Some examples can be run by roles with fewer rights, or different rights. The prerequisite role might include more rights than the minimum subset required.

Hello vCloud: A Simplified RESTful Workflow

2

VMware Cloud Director API clients and VMware Cloud Director servers communicate over HTTPS, exchanging XML representations of VMware Cloud Director API objects.

This simplified example of a RESTful workflow includes requests that discover and deploy a particular vApp, in this case, an FTP server with a connection to the public Internet.

These examples assume that you have access to a catalog that includes a vApp template with certain characteristics and an organization network that supports connections to the public Internet. The workflow and examples are flexible, and can accommodate various vApp templates and cloud capabilities.

Prerequisites

If you want to run the Hello vCloud examples, verify that the following conditions are met.

- You have the login credentials of a user with the predefined vApp Author role or another role that has an equivalent set of rights.
- Your organization contains at least one VDC that has at least one network. For information about creating VDCs and networks, see Chapter 7 Managing an Organization .
- Your organization contains a catalog in which at least one vApp template is available. For information about adding a vApp template to a catalog, see Chapter 4 Provisioning an Organization.

Procedure

1 Logging In

To begin using the API, you request the system to create a `Session` object. In this request, you supply your credentials in an `Authorization` header of the form prescribed by the identity provider that your organization uses. The response includes an authorization token, which you must include in subsequent requests.

2 Find a Catalog and a VDC

Before you can deploy a vApp, you must find a vApp template in one of your organization's catalogs and a VDC in your organization to use for the deployment.

3 Retrieve the Contents of a Catalog

You can make a GET request to a catalog URL to retrieve a list of vApp templates and media images referenced by the catalog.

4 Retrieve a Catalog Item

You can examine the list of items in a catalog to find items of interest based on the values of their `name` and `type` attributes. You must retrieve a catalog item to get a `Description` and a usable reference to the underlying object.

5 Retrieve Deployment Information From the VDC

To deploy your template as a vApp, you must specify an organization VDC to deploy it in and an organization VDC network to connect it to.

6 Deploy the vApp

To create a vApp from a vApp template, you must bind the template's abstract resource requirements, such as network connections, storage resources, memory, and CPU capacity, to appropriate resources in the target VDC. This binding operation is called instantiation.

7 Get Information About a vApp

When you instantiate a vApp template, the server returns the URL of the resulting vApp. You can use this URL with a GET request to retrieve information that you can use to connect to the vApp, modify its configuration, and operate it.

8 Displaying the Virtual Machine Console

After a vApp is powered on, you can retrieve a screen ticket from any of its virtual machines. You use that ticket with the VMware HTML Console SDK to access the virtual machine console from a browser.

9 Undeploy, Power Off, and Delete the vApp

After you undeploy a vApp and power it off, you can use an HTTP DELETE request to delete the vApp object.

10 Log Out

To log out and terminate a VMware Cloud Director API session, delete the `Session` you created when you logged in.

Logging In

To begin using the API, you request the system to create a `Session` object. In this request, you supply your credentials in an `Authorization` header of the form prescribed by the identity provider that your organization uses. The response includes an authorization token, which you must include in subsequent requests.

Every version of the VMware Cloud Director API supported by VMware Cloud Director has a login URL that a client can obtain by making an unauthenticated GET request to the `api/versions` URL. See Retrieve the Login URL and List of Supported API Versions. Because all other VMware Cloud Director API requests must be authenticated, any VMware Cloud Director API workflow must begin with a login request that creates a session and returns an authorization token. The token must be included in subsequent VMware Cloud Director API requests.

Prerequisites

Verify that the following conditions are met:

- You know the type of identity provider that your organization uses. See Create a VMware Cloud Director API Session for more about identity providers, or ask your organization administrator.
- You have the login credentials of a user with the predefined vApp Author role or another role that has an equivalent set of rights.
- Your organization contains at least one VDC and one network. For more information about setting up an organization to support the Hello vCloud workflow, see Chapter 7 Managing an Organization .
- Your organization contains a catalog in which at least one vApp template is available. For more information about adding a vApp template to a catalog, see Chapter 4 Provisioning an Organization.

Procedure

1. Make an API `versions` request to VMware Cloud Director to obtain the VMware Cloud Director API login URL.

   This request has the following form:

   ```
   GET http://vcloud.example.com/api/versions
   ```

2. POST a request to the login URL to create a login session.

   Depending on the type of identity provider your organization uses, the form of this request changes.

   - For system administrators:

     ```
     POST https://vcloud.example.com/cloudapi/1.0.0/sessions/provider
     ```

   - For tenants:

     ```
     POST https://vcloud.example.com/cloudapi/1.0.0/sessions
     ```

   See Login Session Request and Response .

3 Examine the response.

A successful request returns an authorization token, which you must include in subsequent VMware Cloud Director API requests.

Example: Login Session Request and Response

To create a session object, you supply your credentials in an `Authorization` header of the form prescribed by the identity provider that your organization uses, then POST a request to the VMware Cloud Director API login URL. This request does not have a body. All the information required to create a session is included in the `Authorization` header. When using OAuth, the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` is `Bearer` and the constructed header must be in the following form.

```
Authorization: Bearer token
```

For more information and other examples of this request for all supported types of identity provider, see Create a VMware Cloud Director API Session.

toke

The `Session` response to a successful login request includes an X-VMWARE-VCLOUD-ACCESS-TOKEN header whose value is an encoded key that you can use, with the value of the X-VMWARE-VCLOUD-TOKEN-TYPE header, to construct an `Authorization` header to include in subsequent requests.

The `Session` response includes `Link` elements that reference the VMware Cloud Director API objects to which you have access rights. This example shows a response for a user named HelloUser who is a member of an organization named ExampleOrg.

Response:

```
200 OK
X-VMWARE-VCLOUD-ACCESS-TOKEN: cn9u.../6bM=
X-VMWARE-VCLOUD-TOKEN-TYPE: Bearer
Content-Type: application/vnd.vmware.vcloud.session+xml;version=30.0
...
<Session
   xmlns="http://www.vmware.com/vcloud/v1.5"
   user="HelloUser"
   org="ExampleOrg"
   ... >
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.orgList+xml"
      href="https://vcloud.example.com/api/org"/>
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.queryList+xml"
      href="https://vcloud.example.com/api/query" />
   <Link
```

```
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/" />
</Session>
```

The response code indicates whether the request succeeded, or how it failed.

- If the request is successful, the server returns HTTP response code 200 (OK) and headers that include:

  ```
  X-VMWARE-VCLOUD-ACCESS-TOKEN: token
  X-VMWARE-VCLOUD-TOKEN-TYPE: type
  ```

  Use the values of these headers to construct an `Authorization` header to use in subsequent VMware Cloud Director API requests. For example, if the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` is `Bearer`, then the constructed header would have this form:

  ```
  Authorization Bearer token
  ```

- If the `Authorization` header is missing from the request, the server returns HTTP response code 403.
- If the credentials supplied in the `Authorization` header are invalid, the server returns HTTP response code 401.

Important The authorization token expires after a configurable interval of client inactivity. The default interval is 30 minutes. After the token expires, you must log in again to obtain a new token. The system administrator can change this default.

Find a Catalog and a VDC

Before you can deploy a vApp, you must find a vApp template in one of your organization's catalogs and a VDC in your organization to use for the deployment.

After you log in, you can make a GET request to your organization's URL to retrieve the XML representation of the organization. This representation shows the organization's attributes and contents, including links to its catalogs and VDCs.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Examine the list of organizations to which you have access.

Make a GET request to the URL in the `href` value of the `orgList` link, which is present in the response to all login requests.

```
GET https://vcloud.example.com/api/org/
```

Unless you are a system administrator, the response to this request is an `OrgList` element containing a single `Org` element, which represents your organization.

```
<OrgList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.vcloud.orgList+xml"
   href="https://vcloud.example.com/api/org">
   <Org
      type="application/vnd.vmware.vcloud.org+xml"
      name="ExampleOrg"
      href="https://vcloud.example.com/api/org/5" />
</OrgList>
```

2 Retrieve the representation of your organization.

See the request portion of Retrieve the Contents of an Organization.

3 Examine the response to find the links to the organization's catalogs and VDCs.

See the response portion of Retrieve the Contents of an Organization.

Example: Retrieve the Contents of an Organization

This example retrieves the `ExampleOrg` organization listed in the `OrgList` element shown in Step 1.

Request:

```
GET https://vcloud.example.com/api/org/5
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.org+xml
...
<Org
   name="ExampleOrg"
   type="application/vnd.vmware.vcloud.org+xml"
   href="https://vcloud.example.com/api/org/5">
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32"
      name="ExampleCatalog" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"
      name="ExampleVdc01" />
   <Link ... />
   <Link ... />
   <Description>Example Corp's Primary Organization</Description>
</Org>
```

Links in the response whose `rel` attribute has a value of `down` are references to objects that the organization contains. This example shows the subset of those items that we reference in the Hello vCloud example:

- A catalog named `ExampleCatalog`, at URL https://vcloud.example.com/api/catalog/32, where you can look for vApp templates.
- An organization VDC named `ExampleVdc01`, at URL https://vcloud.example.com/api/vdc/5, where you can deploy the vApp.

Retrieve the Contents of a Catalog

You can make a GET request to a catalog URL to retrieve a list of vApp templates and media images referenced by the catalog.

To use a vApp template or media image listed in a catalog, retrieve the catalog to discover the set of `CatalogItem` elements it contains, then make an additional request to retrieve the `CatalogItem` of interest.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of your organization.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/org/5
   ```

2 Examine the response to find the links to the organization's catalogs.

   These links have the following form:

   ```
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/id"
      name="catalog_name" />
   ```

3 Retrieve the contents of the catalog.

   Use a GET request of the form shown in the request portion of Retrieve the Contents of a Catalog.

Example: Retrieve the Contents of a Catalog

This example retrieves the catalog shown in the response portion of Retrieve the Contents of an Organization.

Request:

```
GET https://vcloud.example.com/api/catalog/32
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.catalog+xml
...
<Catalog
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleCatalog"
   type="application/vnd.vmware.vcloud.catalog+xml"
   href="https://vcloud.example.com/api/catalog/32">
   <Description>Main Org Catalog</Description>
   <CatalogItems>
      <CatalogItem
         type="application/vnd.vmware.vcloud.catalogItem+xml"
         name="Ubuntu Template with vsftpd"
         href="https://vcloud.example.com/api/catalogItem/221"/>
       <CatalogItem ... />
       <CatalogItem ... />
   </CatalogItems>
</Catalog>
```

Retrieve a Catalog Item

You can examine the list of items in a catalog to find items of interest based on the values of their `name` and `type` attributes. You must retrieve a catalog item to get a `Description` and a usable reference to the underlying object.

Every vApp template or media image that is added to the catalog is represented as a `CatalogItem` element. When a client browses a catalog, it can read only the `name`, `type`, and `href` of each `CatalogItem`. To retrieve an item from the catalog, the client requires more information. In Retrieve a Catalog Item, the client makes a GET request to the URL in the value of the `href` attribute of a `CatalogItem`. The response provides more information, including a description of the referenced object and another URL that the client can use to retrieve a representation of the object.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the representation of a catalog in your organization.

Use a request like this one:

```
GET https://vcloud.example.com/api/catalog/32
```

2 Examine the response to find the `CatalogItem` elements that the catalog contains.

The value of the `name` attribute of a `CatalogItem` element is taken from the `name` attribute of the referenced object. You can use it as a preliminary indicator of what the item represents.

3 Retrieve a `CatalogItem`.

Use a GET request of the form shown in the request portion of Retrieve a Catalog Item.

Example: Retrieve a Catalog Item

This example retrieves the `CatalogItem` shown in the response portion of Retrieve the Contents of a Catalog.

Request:

```
GET https://vcloud.example.com/api/catalogItem/221
```

In addition to the `name` attribute and `Description` element, the `CatalogItem` contains a `rel="up"` link to the catalog that contains it, and other links that you can use to manage the `CatalogItem`.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<CatalogItem
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Ubuntu Template with vsftpd"
   id="urn:vcloud:catalogitem:221"
   href="https://vcloud.example.com/api/catalogItem/221" ... >
  <Link
      rel="up"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/catalogItem/221/metadata" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.catalogItem+xml"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Description>Approved template for public FTP sites</Description>
```

```
    <Entity
       href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
       type="application/vnd.vmware.vcloud.vAppTemplate+xml"
       name="Ubuntu Template with vsftpd"/>
</CatalogItem>
```

Retrieve Deployment Information From the VDC

To deploy your template as a vApp, you must specify an organization VDC to deploy it in and an organization VDC network to connect it to.

Instantiation, deployment, and operation of a vApp all take place in the context of an organization VDC. The XML representation of a VDC object defines that context in detail. For this exercise, you need several pieces of information from the VDC:

- The URL that a client can use to request an `instantiateVAppTemplate` operation in the VDC.
- A list of networks in the organization VDC that the vApp can connect to.

Deployment Information in a VDC shows this subset of VDC contents.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the representation of your organization.

   See the request portion of Retrieve the Contents of an Organization.

2 Examine the `Org` response to find the links to the organization's VDCs.

   Links to VDCs have the form:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.vdc+xml"
   href="https://vcloud.example.com/api/vdc/id"
   name="VDC_name" />
```

3 Retrieve the contents of the VDC.

   Use a GET request of the form shown in the request portion of Deployment Information in a VDC.

Example: Deployment Information in a VDC

This example shows a request to retrieve the XML representation of a VDC. It shows only the subset of the response that contains deployment information.

Request:

```
GET https://vcloud.example.com/api/vdc/5
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vdc+xml
...
<Vdc
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleVdc01"
   type="application/vnd.vmware.vcloud.vdc+xml"
   href="https://vcloud.example.com/api/vdc/5">
   ...
   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.instantiateVAppTemplateParams+xml"
      href="https://vcloud.example.com/api/vdc/5/action/instantiateVAppTemplate" />
   ...
   <AvailableNetworks>
      <Network
         href="https://vcloud.example.com/api/network/14"
         type="application/vnd.vmware.vcloud.network+xml"
         name="Isolated" />
      <Network
         href="https://vcloud.example.com/api/network/54"
         type="application/vnd.vmware.vcloud.network+xml"
         name="Internet" />
   </AvailableNetworks>
   ...
</Vdc>
```

The information that you need is available in the following elements of the response:

- A `Link` element that contains an `action` URL for `instantiateVAppTemplate`. The `rel` attribute of this link has a value of `add`. It implements an action that adds an object (a vApp) to the VDC.
- A list of `AvailableNetworks` that includes all the networks in the VDC.

Deploy the vApp

To create a vApp from a vApp template, you must bind the template's abstract resource requirements, such as network connections, storage resources, memory, and CPU capacity, to appropriate resources in the target VDC. This binding operation is called instantiation.

To deploy the vApp, you construct an `InstantiateVAppTemplateParams` element that specifies a vApp template to use and a network to connect to, then POST the element to the `action/instantiateVAppTemplate` URL of the VDC.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the vApp template.

   Make a GET request to the URL provided in the `href` attribute of the `Entity` contained by the `CatalogItem` that references the template. You can also use the query service to return a list of references to vApp templates that you can access.

2 Examine the template to find the `Vm` elements of the virtual machines that it contains.

   Look for a `NetworkConnection` element in the `Vm`. You need some of the information in that element to create a vApp network that the virtual machine can connect to.

3 Create an `InstantiateVAppTemplateParams` element.

   See Deploying a vApp for guidelines.

4 Make a POST request to the `action/instantiateVAppTemplate` URL of the VDC.

   Supply the `InstantiateVAppTemplateParams` element as the request body.

Results

The server takes the requested action and returns a `VApp` element. The element has a `status` attribute value of `0`, meaning it is unresolved because the vApp is still being constructed. It also contains a `Task` element that tracks the progress of the request.

See the response portion of Deploying a vApp.

Example: Deploying a vApp

This simple `instantiateVAppTemplate` request assumes that the vApp template includes one `Vm` and has no special requirements other than connecting that `Vm` to a network. For a look at a more complex instantiation request, see Instantiate a vApp Template and Modify Virtual Machine Name, Description, and Storage Profile. The `InstantiateVAppTemplateParams` includes the following information:

- A name for the vApp, supplied in the `name` attribute of the `InstantiateVAppTemplateParams` element. This request also provides a description, which is optional but a good practice.
- A reference to a template, obtained from the `href` attribute of the `Entity` contained by the `CatalogItem` that you retrieved in Retrieve a Catalog Item and suppled in the `Source` element of the `InstantiateVAppTemplateParams`.

- Configuration parameters for a vApp network, supplied in the `NetworkConfigSection` element. This specification includes the following parameters:

  - A name for the network, supplied in the `name` attribute of the `NetworkConfigSection` element. The name you specify for the vApp network must match the value of the `network` attribute of the `NetworkConnection` of the `Vm`. This example assumes that this `NetworkConnection` element includes the following values, which specify that the `Vm` connects to a network named `vAppNetwork`:

    ```
    <NetworkConnectionSection
       ...
        <NetworkConnection
          network="vAppNetwork">
          ...
       </NetworkConnection>
    </NetworkConnectionSection>
    ```

  - A reference to the organization VDC network to which the vApp network connects, specified in the `ParentNetwork` element. The URL used in this reference is one shown in the `AvailableNetworks` element in Deployment Information in a VDC.

  - A fence mode, specified in the `FenceMode` element. A value of `bridged` indicates that the vApp network is connected directly to the organization VDC network.

  For more information about creating networks with the VMware Cloud Director API, see About VMware Cloud Director Networks.

The target of the request is the `instantiateVAppTemplate` URL of this VDC. See Deployment Information in a VDC. Because the operation creates a new vApp object, the HTTP request type is POST.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/instantiateVAppTemplate
Content-Type: application/vnd.vmware.vcloud.instantiateVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Linux FTP server"
   deploy="true"
   powerOn="true"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Example FTP Server</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks
         </ovf:Info>
         <NetworkConfig
            networkName="vAppNetwork">
            <Configuration>
               <ParentNetwork
```

```
                href="https://vcloud.example.com/api/network/54" />
               <FenceMode>bridged</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
   </InstantiationParams>
   <Source
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111" />
</InstantiateVAppTemplateParams>
```

The response to the instantiation request is a sparsely populated `vApp` element that includes the following information:

- The status of the vApp. The `status` value `0` indicates that the vApp is unresolved, because instantiation is not complete.
- The name of the vApp, as supplied in the request.
- The vApp URL, shown in the `href` attribute of the `VApp` element. You can use this reference to retrieve information about the vApp.
- A task created to track the instantiation. The `Task` element has an `operation` attribute that describes what is happening, and contains an `Owner` element that is a reference the vApp being created. The vApp is the owner of the task.

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   deployed="false"
   status="0"
   name="Linux FTP server"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-7">
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"/>
   <Description>Example FTP Server vApp</Description>
   <Tasks>
      <Task
         status="queued"
         operation="Creating Virtual Application Linux FTP server(7)"
         ... >
         <Owner
            type="application/vnd.vmware.vcloud.vApp+xml"
            name="LinuxFtpServer"
            href="https://vcloud.example.com/vApp/vapp-7" />
      </Task>
   </Tasks>
</VApp>
```

Note The `Task` has an initial `status` of `queued`. When the `Task` is able to run, the `status` changes to `running` and then `completed`. You can GET the `Task` to follow the status as it changes. Elapsed time between the task being queued and starting to run is affected by the number of tasks already in the queue, and by system settings that limit resource-intensive operations.

Get Information About a vApp

When you instantiate a vApp template, the server returns the URL of the resulting vApp. You can use this URL with a GET request to retrieve information that you can use to connect to the vApp, modify its configuration, and operate it.

As other examples have shown, a client can always use an HTTP GET request to the URL in the object's `href` attribute to discover the current state of any VMware Cloud Director API object, including a vApp.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1. Retrieve the XML representation of the vApp.

   Make a GET request to the URL in the `href` attribute of the `VApp` element that is returned when you create the vApp from the template.

2. Examine the response.

   See Getting Information About the vApp.

Example: Getting Information About the vApp

This response reveals several things about the vApp:

- The vApp is deployed (its `deployed` attribute is set to `true`) and powered on (`status="4"`). See Object Creation Status.
- The `Vm` in its `Children` collection is also powered on and deployed. The `Vm` is connected to the vApp network created during instantiation. See Deploying a vApp. Properties of this network are included in the `NetworkConfigSection` of the vApp, although most are not shown here. Properties of the virtual machine's connection to the network, including its IP address, are shown in the `NetworkConnection` of the `Vm`.

- Action links for all operations except `powerOn` are present in the `VApp` element and the `Vm` element that it contains. Because the vApp is already powered on, that operation is invalid for the vApp in its current state, so the link is not part of the response. The link for `deploy` is always present, even in a deployed vApp, because the `deploy` action is always valid. The `Vm` element also includes several links for actions that are not applicable to a vApp. Actions such as acquiring a screen ticket or thumbnail, and inserting or removing media, are meaningful only in the context of a virtual machine. Other actions, like shutdown and reboot, can be applied to either object. See Chapter 5 Deploying and Operating vApps and Virtual Machines .

Request:

```
GET https://vcloud.example.com/api/vApp/vapp-7
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
   ...
   deployed="true"
   status="4"
   name="Linux FTP server"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-7" ... >
   ...
   <Link
      rel="power:reboot"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/reboot" />
   <Link
      rel="power:powerOff"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/powerOff" />
   <Link
      rel="undeploy"
      href="https://vcloud.example.com/api/vApp/vapp-7/action/undeploy" />
   <Link
      rel="deploy"
      href="https://vcloud.example.com/api/vApp/vapp-7/action/deploy" />
   <Link
      rel="power:shutdown"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/shutdown" />
   <Link
      rel="power:reset"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/reset" />
   <Link
      rel="power:suspend"
      href="https://vcloud.example.com/api/vApp/vapp-7/power/action/suspend" />
   <Link ... />
   ...
   <Description>Example FTP Server vApp</Description>
   <LeaseSettingsSection ... >
      ...
   </LeaseSettingsSection>
   <ovf:StartupSection ... >
      ...
   </ovf:StartupSection>
   <ovf:NetworkSection ... >
      <ovf:Info />
      <ovf:Network
         ovf:name="vAppNetwork">
         <ovf:Description />
      </ovf:Network>
   </ovf:NetworkSection>
   <NetworkConfigSection
      href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/"
      ovf:required="false">
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.networkConfigSection+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/" />
      <ovf:Info>Configuration parameters for vAppNetwork</ovf:Info>
      <NetworkConfig
         networkName="vAppNetwork">
         <Configuration>
            <IpScopes>
               ...
            </IpScopes>
            <ParentNetwork
               type="application/vnd.vmware.vcloud.network+xml"
               name="Internet"
               href="https://vcloud.example.com/api/network/54" />
            <FenceMode>bridged</FenceMode>
         </Configuration>
         <IsDeployed>true</IsDeployed>
      </NetworkConfig>
   </NetworkConfigSection>
   <Children>
      <Vm
         deployed="true"
         status="4"
         name="ubuntu10-x86"
         type="application/vnd.vmware.vcloud.vm+xml"
         href="https://vcloud.example.com/api/vApp/vm-4">
         ...
         <Link
            rel="power:reboot"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/reboot" />
         <Link
            rel="power:powerOff"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/powerOff" />
         <Link
            rel="undeploy"
            href="https://vcloud.example.com/api/vApp/vm-4/action/undeploy" />
         <Link
            rel="deploy"
            href="https://vcloud.example.com/api/vApp/vm-4/action/deploy" />
         <Link
```

```
            rel="power:shutdown"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/shutdown" />
         <Link
            rel="power:reset"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/reset" />
         <Link
            rel="power:suspend"
            href="https://vcloud.example.com/api/vApp/vm-4/power/action/suspend" />
         <Link
            rel="up"
            type="application/vnd.vmware.vcloud.vApp+xml"
            href="https://vcloud.example.com/api/vApp/vapp-7" />
         <Link
            rel="screen:thumbnail"
            href="https://vcloud.example.com/api/vApp/vm-4/screen" />
         <Link
            rel="screen:acquireTicket"
            href="https://vcloud.example.com/api/vApp/vm-4/screen/action/acquireTicket" />
         <Link
            rel="media:insertMedia"
            type="application/vnd.vmware.vcloud.mediaInsertOrEjectParams+xml"
            href="https://vcloud.example.com/api/vApp/vm-4/media/action/insertMedia" />
         <Link ... />
         ...
         <Description />
         <ovf:VirtualHardwareSection>
            ...
         </ovf:VirtualHardwareSection>
         ...
         <NetworkConnectionSection>
            ...
            <NetworkConnection>
               network="vAppNetwork">
               <NetworkConnectionIndex>0</NetworkConnectionIndex>
               <IpAddress>10.147.201.10</IpAddress>
               <IsConnected>true</IsConnected>
               <MACAddress>00:50:56:01:01:49</MACAddress>
               <IpAddressAllocationMode>DHCP</IpAddressAllocationMode>
            </NetworkConnection>
         </NetworkConnectionSection>
         <GuestCustomizationSection>
            ...
         </GuestCustomizationSection>
         ...
      </Vm>
   </Children>
</VApp>
```

Displaying the Virtual Machine Console

After a vApp is powered on, you can retrieve a screen ticket from any of its virtual machines. You use that ticket with the VMware HTML Console SDK to access the virtual machine console from a browser.

Each `Vm` element in a vApp includes a link where `rel="screen:acquireMksTicket"` if the virtual machine it represents is powered on. You can use that link to retrieve a screen ticket that includes all the information you need to create a VMware HTML Console to access the virtual machine.

Note A `Vm` element might also contain a link of the form:

```
<Link
   rel="screen:acquireTicket"
   href="https://vcloud.example.com/api/vApp/vm-4/screen/action/acquireTicket" />
```

This link is provided for compatibility with older systems. The ticket returned by a request to this link is not compatible with this release of VMware Cloud Director.

Prerequisites

- This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.
- Verify that the virtual machine whose console you want to display is powered on.
- Download and unzip the VMware HTML Console SDK version 2.1 or later.

Procedure

1 Retrieve the screen ticket.

POST a request to the `acquireMksTicket` link of the `Vm`.

Request:

```
POST https://vcloud.example.com/api/vApp/vm-4/screen/action/acquireMksTicket
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.mksticket+xml
...
<MksTicket
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   ... >
   <Host>10.150.130.63</Host>
   <Vmx></Vmx>
   <Ticket>ticket-string</Ticket>
   <Port>902</Port>
</MksTicket>
```

2 Construct an HTML `Script` element from the information in the `MksTicket` response. Skip the `<Vmx>` element.

See Using an MksTicket in a VMware HTML Console SDK Script.

3 Use the `Script` element with the VMware HTML Console SDK .

Replace the final `script` element in the Quick Start example shown in the *VMware HTML Console SDK Release Notes* (release number 2.1, build number 4504321) with the script you constructed. Replace the *ticket-string* with one from a freshly-retrieved `MksTicket`, then open the example in a browser.

Example: Using an `MksTicket` in a VMware HTML Console SDK Script

This HTML fragment shows a `Script` element derived from an example in the VMware HTML Console SDK *Release Notes*. In this example:

- The URL passed as a parameter to the SDK `wmks.connect` method includes the following information from the `MksTicket`.
  - *console-ip-address* is the IP address of the virtual machine, returned in the `Host` element of the `MksTicket`.
  - *console-port* is the VMware Cloud Director console proxy port (defaults to 443).
  - *Mks-port* is the value returned in the `Port` element of the `MksTicket`
  - *ticket* is the string returned in the `Ticket` element of the `MksTicket`. The *ticket-string* is valid for 30 seconds from the time you retrieve the `MksTicket`.

```
<script>
var wmks = WMKS.createWMKS("wmksContainer",{VCDProxyHandshakeVmxPath:
        "vmfs/volumes/5331e00b-467faf9c-5561-d48564677c70/example-vm (19115346-c01c-4c9b-a21f-
d487865a9f98)/example-vm
       (19115346-c01c-4c9b-a21f-d487865a9f98).vmx.vmx", enableUint8Utf8:true})
 .register(WMKS.CONST.Events.CONNECTION_STATE_CHANGE, function(event,data){
 if(data.state == WMKS.CONST.ConnectionState.CONNECTED){
  console.log("connection state change : connected");}
 });
wmks.connect("wss://10.150.130.63:443/902;ticket-string");
</script>
```

Undeploy, Power Off, and Delete the vApp

After you undeploy a vApp and power it off, you can use an HTTP DELETE request to delete the vApp object.

A deployed vApp has a link that you can use with a POST request to undeploy the vApp and take a power action such as powering it off or suspending it. A powered-off vApp has a link that you can use with a DELETE request to remove the vApp.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the vApp.

Make a GET request to the URL in the `href` attribute of the `VApp` element that was returned when you created the vApp from the template. See Get Information About a vApp.

2 Undeploy the vApp, and specify that it should also be powered off.

Make a POST request to the vApp `action/undeploy` link, which has the following form:

```
<Link
   rel="undeploy"
   href="https://vcloud.example.com/api/vApp/vapp-7/action/undeploy"/>
```

In the request body, specify that the undeployment include powering off the vApp. See Undeploy, Power Off, and Delete a vApp.

3 Retrieve the XML representation of the vApp again.

After it is powered off and undeployed, the vApp includes a `rel="remove"` link of the following form:

```
<Link
   rel="remove"
   href="https://vcloud.example.com/api/vApp/vapp-7"/>
```

4 Remove the vApp.

Make a DELETE request to the vApp's `rel="remove"` link, as shown in the request portion of Undeploy, Power Off, and Delete a vApp.

Results

The server starts a task to manage the events that lead up to the removal of the vApp, and returns a `Task` element that you can use to track the progress of the task.

Example: Undeploy, Power Off, and Delete a vApp

You can use the `undeploy` request body, an `UndeployVAppParams` element, to specify an `UndeployPowerAction` element. This example specifies an `UndeployPowerAction` of `powerOff`. Even though `powerOff` is the default `UndeployPowerAction`, It appears here for completeness.

Request:

```
POST https://vcloud.example.com/api/vApp/vapp-7/action/undeploy
Content-Type: application/vnd.vmware.vcloud.undeployVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<UndeployVAppParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <UndeployPowerAction>powerOff</UndeployPowerAction>
</UndeployVAppParams>
```

Response:

```
202 Accepted
...
<Task
   xmlns="http://www.vmware.com/vcloud/v1.5"
   ...
   operation="Undeploying Virtual Application Linux FTP server (7)"
   ... >
</Task>
```

After the vApp is undeployed and powered off, its representation includes a link where `rel="remove"`. Make a DELETE request to this link to remove the vApp.

Request:

```
DELETE https://vcloud.example.com/api/vApp/vapp-7
```

Response:

```
202 Accepted
...
<Task
   ...
   operation="Deleting Virtual Application Linux FTP server (7)"
   ... >
</Task>
```

Note You can force deletion of a vApp that is powered on, deployed, or both by appending a `force=true` query string to the vApp `href` and using that URL with a DELETE request, as shown here:

```
DELETE https://vcloud.example.com/api/vApp/vapp-7?force=true
```

Log Out

To log out and terminate a VMware Cloud Director API session, delete the `Session` you created when you logged in.

The logout request, like all other authenticated requests, must include the authorization header, as shown in Logging Out.

Prerequisites

Verify that you are logged in.

Procedure

◆ Make a DELETE request specifying the `href` of the current `Session` object.

Example: Logging Out

This example deletes the current user's `Session`, which logs the user out.

Request:

```
DELETE https://vcloud.example.com/api/session
```

Response:

```
204 No Content
```

Exploring a Cloud

3

You can use HTTP GET requests to browse containers such as organizations, catalogs, and VDCs in a cloud.

Responses to these requests include metadata about the container itself and references to the objects it contains. These references are provided in `Link` elements, which have `href` attributes whose values the client can use in requests to get more information about the objects themselves. This process is sometimes called serial discovery, because the contents of one response provides links to locations where you can look for more information. The hierarchical structure of VMware Cloud Director API container objects lends itself to graphical representation as a folder hierarchy or tree view of VMware Cloud Director API objects, and enables clients to use the same set of objects and operations to implement a breadth-first or depth-first approach to browsing.

The list of entry points from which you can begin browsing is contained in the `Session` element that is returned in response to a successful login. The contents of this list is based on your role and privileges.

Read the following topics next:

- Create a VMware Cloud Director API Session
- Retrieve a List of Organizations Accessible to You
- Retrieve an Administrative View of a Cloud
- Retrieve a List of vSphere Platform Operations and Objects for a Cloud

Create a VMware Cloud Director API Session

The VMware Cloud Director API login mechanism authenticates a user and creates a `Session` object that contains the URLs from which that user can begin browsing objects in an organization. Users must submit their credentials in the form required by the identity provider that their organization specifies.

The VMware Cloud Director API does not have a login request. To begin using the API, you request the system to create a `Session` object, supplying your credentials in an `Authorization` header of the form prescribed by the identity provider that your organization uses. The response includes a JSON Web Token (JWT), as defined in RFC 7519, and a `Session` element whose `Link` elements reference the VMware Cloud Director API objects to which you have access rights.

VMware Cloud Director supports several forms of authentication to the VMware Cloud Director API.

- OAuth authentication, as defined in RFC 6749. Users defined in an OAuth identity provider must acquire an OAuth token from that identity provider and include it in the request to create a VMware Cloud Director API `Session`.
- SAML (Security Assertion Markup Language) authentication, as defined in RFC 6595. Users defined in a SAML identity provider must acquire and process a security assertion from that identity provider and include the processed assertion and other attributes in the request to create a VMware Cloud Director API `Session`.
- Basic HTTP authentication, as specified in RFC 2617. LDAP users and local users are defined by the VMware Cloud Director integrated identity provider, and must include credentials in the form required by Basic HTTP authentication in the request to create a VMware Cloud Director API `Session`.

All organizations are created with implicit support for the VMware Cloud Director integrated identity provider. Organizations can be configured to add support for a SAML or OAUTH identity provider.

Important Regardless of the authentication protocol your organization uses, the JWT token returned with a `Session` object is supported for use in the `Authorization` header of all requests, and is required for requests that span associations of organizations or sites. See Configuring and Managing Multisite Deployments.

A `Session` object can be deleted by its owner or an administrator. After your `Session` expires or is deleted, you are not authenticated.

A `Session` object expires after a configurable interval of client inactivity. To change the length of this client inactivity timeout, a system administrator can change the value of `SessionTimeoutMinutes` in the system's `GeneralSettings`. See Retrieve or Update System Settings.

All requests to create a login session include a similar set of steps.

Procedure

1. Make an API `versions` request to retrieve the list of supported API versions and the login URL for each version.

   The request has this form:

   ```
   GET http://vcloud.example.com/api/versions
   ```

   You do not need to be authenticated to make a `versions` request.

2. POST a request to the login URL, supplying your credentials in the request's Authorization header.

   These requests have the following form.

   - For system administrators:

     ```
     POST https://vcloud.example.com/cloudapi/1.0.0/sessions/provider
     Authorization: auth-type credentials
     Accept: application/*;version=api-version
     ```

   - For tenants:

     ```
     POST https://vcloud.example.com/cloudapi/1.0.0/sessions
     Authorization: auth-type credentials
     Accept: application/*;version=api-version
     ```

   The value of *auth-type* depends on the protocol you are using. Each authentication protocol has its own requirements for credentials and other attributes.

   Basic

   Specifies Basic HTTP authentication. Requires your user name, organization name, and password in a MIME Base64 encoding.

   Bearer

   Used with OAuth authentication. Requires an OAuth token and your organization name.

   Sign

   Used with SAML authentication. Requires a compressed, encoded SAML assertion and other attributes.

3. Examine the response to retrieve the authorization token and links to objects that you have rights to access.

   A successful request returns a `Session` element and headers that include:

   ```
   X-VMWARE-VCLOUD-ACCESS-TOKEN: token
   X-VMWARE-VCLOUD-TOKEN-TYPE: type
   ```

   Use the values of these headers to construct a JWT `Authorization` header to use in subsequent VMware Cloud Director API requests. For example, if the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` is `Bearer`, then the constructed JWT header would have this form:

   ```
   Authorization Bearer token
   ```

Create a Session Using OAuth Authentication

Users defined in an organization that specifies an OAuth identity provider must acquire an OAuth token from the identity provider and include it in the request to create a `Session`.

Prerequisites

- Verify that you know the API login URL. See Retrieve the Login URL and List of Supported API Versions
- Verify that you are logging in as a user whose identity is managed by the OAuth identity provider defined by your organization.

Procedure

1. Acquire the OAuth token from your identity provider.
2. Use the login URL to authenticate to the VMware Cloud Director API.

   POST a request to this URL. The request must include an `Authorization` header that specifies `Bearer` as the authorization method, includes an OAuth token retrieved from your identity provider, and has the following attributes:

   Table 3-1. OAuth Authorization Header Attributes and Values

Attribute Name	Attribute Value
`org`	The name of your VMware Cloud Director organization.

   See OAuth Login Request and Response.
3. Examine the response.

   The response code indicates whether the request succeeded, or how it failed.

   - If the request is successful, the server returns HTTP response code 200 (OK) and headers that include:

     ```
     X-VMWARE-VCLOUD-ACCESS-TOKEN: token
     X-VMWARE-VCLOUD-TOKEN-TYPE: type
     ```

     Use the values of these headers to construct an `Authorization` header to use in subsequent VMware Cloud Director API requests. For example, if the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` is `Bearer`, then the constructed header would have this form:

     ```
     Authorization Bearer token
     ```
   - If the `Authorization` header is missing from the request, the server returns HTTP response code 403.
   - If the credentials supplied in the `Authorization` header are invalid, the server returns HTTP response code 401.

Results

A valid request returns a `Session` element. See OAuth Login Request and Response

Example: OAuth Login Request and Response

This example shows an OAuth login request and response for a user logging in to the `Finance` organization of a cloud whose API login URL is https://vcloud.example.com/cloudapi/1.0.0/sessions.

The following credentials are required:

OAuth-token

The token returned by your OAuth identity provider.

org

The name of your organization.

Request:

```
POST https://vcloud.example.com/cloudapi/1.0.0/sessions
Authorization: Bearer OAuth-token; org=Finance
Accept: application/*;version=9.0
```

Response:

```
200 OK
...
<Session
   xmlns="http://www.vmware.com/vcloud/v1.5"
   userUrn="urn:vcloud:user:fe50b0b5-..."
   user="bob"
   org="Finance"
   ... >
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.org+xml"
      name="System"
      href="https://vcloud.example.com/api/org/5" />
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.queryList+xml"
      href="https://vcloud.example.com/api/query" />
   <Link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/" />
   <Link
      rel="down:extensibility"
      type="application/vnd.vmware.vcloud.apiextensibility+xml"
      href="https://vcloud.example.com/api/extensibility" />
</Session>
```

The response includes several `Link` types, including:

org

A link to your organization. See Retrieve a List of Organizations Accessible to You.

queryList

A link to the set of typed queries the user can run. See Chapter 10 Using the Query Service.

entity

A link to the entity resolver. See Retrieve an Object as an Entity.

extensibility

A link to the extensibility framework entry point. See VMware Cloud Director Extension Services.

Create a Session Using SAML Authentication

Users defined in an organization that specifies a SAML identity provider must acquire and process a security assertion from that identity provider and include the processed assertion and other attributes in the request to create a VMware Cloud Director API `Session`.

The VMware Cloud Director API login mechanism supports SAML authentication using two types of security assertions:

- Bearer assertions, which make no guarantees about message integrity and claimed client identity.
- Holder-of-key assertions, which guarantee subject identity by including a signature generated with the subject's private key.

Prerequisites

- Verify that you know the API login URL. See Retrieve the Login URL and List of Supported API Versions
- Verify that you are logging in as a user whose identity is managed by the SAML identity provider defined by your organization.

Procedure

1 Acquire the SAML assertion from your organization's identity provider.

2 Compress the assertion using GZIP.

3 Encode the compressed assertion using a MIME Base64 encoding, as specified in RFC 1421.

4 Use the login URL to authenticate to the VMware Cloud Director API.

POST a request to this URL. The request must include an `Authorization` header that specifies `Sign` as the authorization method and has the following attributes:

Table 3-2. SAML Authorization Header Attributes and Values

Attribute Name	Attribute Value
`token`	The compressed, encoded identity assertion from your SAML identity provider.
`signature`	Base64 encoded signature of the `token` XML (the uncompressed identity assertion from your SAML identity provider) generated using the client's private key. Required when using holder-of-key subject confirmation.
`signature_alg`	The algorithm used to generate the `signature`, expressed as one of the values listed in http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#Signature. Required if `signature` is present.
`org`	The name of your VMware Cloud Director organization. Defaults to `org="system"` if not specified.

See Create a Login Session Using a SAML Identity Provider.

5 Examine the response.

The response code indicates whether the request succeeded, or how it failed.

- If the request is successful, the server returns HTTP response code 200 (OK) and headers that include:

```
X-VMWARE-VCLOUD-ACCESS-TOKEN: token
X-VMWARE-VCLOUD-TOKEN-TYPE: type
```

  Use the values of these headers to construct an `Authorization` header to use in subsequent VMware Cloud Director API requests. For example, if the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` is `Bearer`, then the constructed header would have this form:

```
Authorization Bearer token
```

- If the `Authorization` header is missing from the request, the server returns HTTP response code 403.
- If the credentials supplied in the `Authorization` header are invalid, the server returns HTTP response code 401.

Results

A valid request returns a `Session` element. See Create a Login Session Using a SAML Identity Provider

Example: Create a Login Session Using a SAML Identity Provider

This example shows a SAML login request and response for a user logging in to the `Finance` organization of a cloud whose API login URL is https://vcloud.example.com/cloudapi/1.0.0/sessions. This example shows two varieties of the request.

Request (bearer token):

```
POST https://vcloud.example.com/cloudapi/1.0.0/sessions
Authorization: Sign token="compressed-encoded-credentials",
org="Finance"
Accept: application/*;version=9.0
```

When using a SAML assertion that provides holder-of-key (HOK) subject confirmation, the request header must include `signature` and `signature_alg` attributes, as shown in this example, which assumes a signature created with a SHA encoding and RSA encryption algorithms:

Request (holder-of-key token):

```
POST https://vcloud.example.com/cloudapi/1.0.0/sessions
Authorization: Sign token="compressed-encoded-credentials",
org="Finance",
signature="encoded-signature",
signature_alg="SHA256withRSA"
Accept: application/*;version=9.0
```

The response is the same in both cases.

Response:

```
200 OK
...
<Session
   xmlns="http://www.vmware.com/vcloud/v1.5"
   userUrn="urn:vcloud:user:fe50b0b5-..."
   user="bob"
   org="Finance"
   ... >
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.org+xml"
      name="System"
      href="https://vcloud.example.com/api/org/5" />
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.queryList+xml"
      href="https://vcloud.example.com/api/query" />
   <Link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/" />
   <Link
```

```
      rel="down:extensibility"
      type="application/vnd.vmware.vcloud.apiextensibility+xml"
      href="https://vcloud.example.com/api/extensibility" />
</Session>
```

The response includes several `Link` types, including:

org

A link to your organization. See Retrieve a List of Organizations Accessible to You.

queryList

A link to the set of typed queries the user can run. See Chapter 10 Using the Query Service.

entity

A link to the entity resolver. See Retrieve an Object as an Entity.

extensibility

A link to the extensibility framework entry point. See VMware Cloud Director Extension Services.

Create a Session Using Basic Authentication

LDAP users and local users are defined by the VMware Cloud Director integrated identity provider, and must include credentials in the form required by Basic HTTP authentication when making a the request to create a VMware Cloud Director API `Session`.

Prerequisites

- Verify that you know the API login URL. See Retrieve the Login URL and List of Supported API Versions.
- Verify that you are logging in as a user whose identity is managed by the VMware Cloud Director integrated identity provider.

Procedure

1 Use the login URL to authenticate to the VMware Cloud Director API.

POST a request to this URL. The request must include your username, organization name, and password in a MIME Base64 encoding. See Create a Login Session Using the Integrated Identity Provider.

2 Examine the response.

The response code indicates whether the request succeeded, or how it failed.

- If the request is successful, the server returns HTTP response code 200 (OK) and headers that include:

  ```
  X-VMWARE-VCLOUD-ACCESS-TOKEN: token
  X-VMWARE-VCLOUD-TOKEN-TYPE: type
  ```

  Use the values of these headers to construct an `Authorization` header to use in subsequent VMware Cloud Director API requests. For example, if the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` is `Bearer`, then the constructed header would have this form:

  ```
  Authorization Bearer token
  ```

- If the `Authorization` header is missing from the request, the server returns HTTP response code 403.
- If the credentials supplied in the `Authorization` header are invalid, the server returns HTTP response code 401.

Results

A valid request returns a `Session` element. See Create a Login Session Using the Integrated Identity Provider.

Example: Create a Login Session Using the Integrated Identity Provider

A request to create a login session using the Integrated Identity provider must supply the user's credentials in the following form:

```
user@organization:password
```

- *user* is the user's login name.
- *organization* is the name of the user's organization.
- *password* is the user's password.

These credentials must be supplied in a MIME Base64 encoding, as specified in RFC 1421.

This example shows a Basic HTTP authentication login request and response for a user logging in to the `Finance` organization of a cloud whose API login URL is https://vcloud.example.com/cloudapi/1.0.0/sessions.

Request:

```
POST https://vcloud.example.com/cloudapi/1.0.0/sessions
Authorization: Basic encoded-credentials
Accept: application/*;version=9.0
```

Response:

```
200 OK
...
<Session
   xmlns="http://www.vmware.com/vcloud/v1.5"
   userUrn="urn:vcloud:user:fe50b0b5-..."
   user="bob"
   org="Finance"
   ... >
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.org+xml"
      name="System"
      href="https://vcloud.example.com/api/org/5" />
  <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.queryList+xml"
      href="https://vcloud.example.com/api/query" />
   <Link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/" />
   <Link
      rel="down:extensibility"
      type="application/vnd.vmware.vcloud.apiextensibility+xml"
      href="https://vcloud.example.com/api/extensibility" />
</Session>
```

The response includes several `Link` types, including:

org

A link to your organization. See Retrieve a List of Organizations Accessible to You.

queryList

A link to the set of typed queries the user can run. See Chapter 10 Using the Query Service.

entity

A link to the entity resolver. See Retrieve an Object as an Entity.

extensibility

A link to the extensibility framework entry point. See VMware Cloud Director Extension Services.

Retrieve a List of Organizations Accessible to You

A successful login request returns a `Session` element, which contains `Link` elements that reference the organizations that you are permitted to access.

Every authenticated user has an associated `Session` object that contains one or more `Link` elements. The set of `Link` elements in your `Session` is based on your role and privileges. Each of these elements includes a URL that you can use with a GET request to explore a subset of objects in the cloud.

All `Session` elements include a link that you can use to retrieve an `OrgList` element. Unless you are the system administrator, this list includes just the organization to which you logged in. For a system administrator, the list includes all organizations in the cloud.

Prerequisites

Create a login session. See Create a Session Using Basic Authentication.

Procedure

1 Retrieve the XML representation of your `Session` object.

Use a request like this one:

```
GET https://vcloud.example.com/api/session
```

2 Examine the contents of the `Session` element to locate the link to the organization list.

This link has the following form:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.orgList+xml"
   href="https://vcloud.example.com/api/org"/>
```

3 Retrieve the list of organizations by making a GET request to the `href` value of the `Link`.

See Retrieve an Organization List.

Example: Retrieve an Organization List

Request:

```
GET https://vcloud.example.com/api/org
```

The request returns an `OrgList` element similar to the one shown here. Additional `Org` elements are returned only when a system administrator makes the request.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.orgList+xml
...
<OrgList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.vcloud.orgList+xml"
   href="https://vcloud.example.com/api/org">
   <Org
      type="application/vnd.vmware.vcloud.org+xml"
      name="ExampleOrg"
      href="https://vcloud.example.com/api/org/5" />
   <Org ... />
   <Org ... />
</OrgList>
```

Retrieve an Administrative View of a Cloud

A successful login by an organization or system administrator returns a Session element that contains a link that you can use to retrieve a VCloud element. The VCloud element provides access to a cloud-wide namespace of objects that an organization administrator can view and, in most cases, modify.

The primary administrative objects in a cloud include organizations, provider VDCs, rights, roles, and external networks. Each object type is represented in a VCloud element by zero or more references. The VMware Cloud Director API defines several objects that are used only in administrative operations. Some, like User, Group, and Role, are unique to administrative operations. Others extend common VMware Cloud Director API objects to add elements and attributes that only administrators can view or modify. An AdminOrg, for example, provides an administrative view of an Org, and an AdminVdc does the same thing for a Vdc.

A system administrator can obtain more information about any of these objects by making a GET request to its URL, which is the value of its href attribute.

The vCloud element includes links that enable a system administrator to add organizations and roles. Subordinate objects such as users, catalogs, and VDCs are contained by individual organizations and not listed at this level.

Prerequisites

Use the credentials of an organization administrator or system administrator to create a login session. See Create a Session Using Basic Authentication.

Procedure

1 Retrieve the XML representation of your Session object.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/session
   ```

2 Examine the contents of the Session element to locate the link to the VCloud object.

   This link has the following form:

   ```
   <Link
      rel="down"
      type="application/vnd.vmware.admin.vcloud+xml"
      href="https://vcloud.example.com/api/admin"/>
   ```

3 Retrieve the `VCloud` element by making a GET request to the `href` value of the `Link` described in Step 2.

See Retrieve an Administrative View of a Cloud

Example: Retrieve an Administrative View of a Cloud

Request:

```
GET https://vcloud.example.com/api/admin
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vcloud+xml
...
<VCloud
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="vCloud"
   href="https://vcloud.example.com/api/admin">
   <Link
      rel="add"
      type="application/vnd.vmware.admin.role+xml"
      href="https://vcloud.example.com/api/admin/roles" />
   <Link
      rel="add"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/orgs" />
   <Link
      rel="down"
      type="application/vnd.vmware.admin.systemOrganization+xml"
      name="System"
      href="https://vcloud.example.com/api/admin/org/99" />
   <Link ... />
   ...
   <Description>Example Corporation’s vCloud</Description>
   <OrganizationReferences>
     <OrganizationReference
         type="application/vnd.vmware.admin.organization+xml"
         name="Engineering"
         href="https://vcloud.example.com/api/admin/org/1"/>
      <OrganizationReference
         type="application/vnd.vmware.admin.organization+xml"
         name="Engineering"
         href="https://vcloud.example.com/api/admin/org/44"/>
      <OrganizationReference ... />
      ...
   </OrganizationReferences>
   <ProviderVdcReferences>
      <ProviderVdcReference
         type="application/vnd.vmware.admin.providervdc+xml"
         name="Main Provider"
         href="https://vcloud.example.com/api/admin/providervdc/2" />
      <ProviderVdcReference ... />
      ...
   </ProviderVdcReferences>
   <RightReferences>
      <RightReference
         href="https://vcloud.example.com/api/admin/right/4886663f-
ae31-37fc-9a70-3dbe2f24a8c5"
         name="Catalog: Add vApp from My Cloud"
         type="application/vnd.vmware.admin.right+xml"/>
      <RightReference
         href="https://vcloud.example.com/api/admin/right/39ec03d4-440d-32cf-8507-
f01acd822540"
         name="Catalog: Change Owner"
         type="application/vnd.vmware.admin.right+xml"/>
      ...
      <RightReference
         href="https://vcloud.example.com/api/admin/right/94b041c2-04cd-3a9b-
a3ff-0ba57814cff4"
         name="VDC Template: View"
         type="application/vnd.vmware.admin.right+xml"/>
   </RightReferences>
   <RoleReferences>
      <RoleReference
         href="https://vcloud.example.com/api/admin/role/67e119b7-083b-349e-8dfd-6cf0c19b83cf"
         name="System Administrator"
         type="application/vnd.vmware.admin.role+xml"/>
         ...
   </RoleReferences>
   <Networks>
      <Network
         type="application/vnd.vmware.admin.network+xml"
         name="ExternalNetwork-VC1"
         href="https://vcloud.example.com/api/admin/network/7" />
      <Network
         type="application/vnd.vmware.admin.network+xml"
         name="ExternalNetwork-VC2"
         href="https://vcloud.example.com/api/admin/network/33" />
    </Networks>
</VCloud>
```

Retrieve a List of vSphere Platform Operations and Objects for a Cloud

A successful login by a system administrator returns a `Session` element that contains a link that you can use to retrieve a `VMWExtension` element.

Every VMware Cloud Director installation depends on vSphere platform resources such as vCenter serversvCenter Server instances and hosts, vShield Manager, portgroups, virtual switches, and so on. The `VMWExtension` element provides access to a cloud-wide namespace of vSphere platform objects that are registered for use by the system, and links that allow you to add vSphere servers and related resources to your cloud. Objects in the `admin/extension` XML namespace provide a system administrator with programmatic access to these resources.

Prerequisites

Use the credentials of a system administrator to create a login session. See Create a Session Using Basic Authentication.

Procedure

1 Retrieve the XML representation of your `Session` object.

Use a request like this one:

```
GET https://vcloud.example.com/api/session
```

2 Examine the contents of the `Session` element to locate the link to the `VMWExtension` object.

This link has the following form:

```
<Link
   rel="down"
   type="application/vnd.vmware.admin.vmwExtension+xml"
   href="https://vcloud.example.com/api/admin/extension"/>
```

3 Retrieve the list of organizations by making a GET request to the `href` value of the `Link` described in Step 2.

The request returns a `VMWExtension` element, as shown in Retrieve a List of vSphere Platform Operations and Objects for a Cloud.

Example: Retrieve a List of vSphere Platform Operations and Objects for a Cloud

Request:

```
GET https://vcloud.example.com/api/admin/extension
```

The response is a `VMWExtension` element containing a number of `Link` elements. This example shows only a subset of `VMWExtension` contents.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vmwextension+xml
...
<vmext:VMWExtension
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.vmwExtension+xml">
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwProviderVdcReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/providerVdcReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwExternalNetworkReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/externalNetworkReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwVimServerReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServerReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwHostReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/hostReferences" />
   <vcloud:Link ... />
   ...
  <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.extensionServices+xml"
      href="https://vcloud.example.com/api/admin/extension/service" />
</vmext:VMWExtension>
```

Provisioning an Organization

4

The VMware Cloud Director API provides several ways for you to make vApp templates, vApps, media images, and independent disks available to users in a VMware Cloud Director organization.

The VMware Cloud Director API allows you to upload and download OVF packages and ISO-format media images. Operations are characterized as uploads when they transfer content from a VMware Cloud Director API client system to a target catalog in a VMware Cloud Director organization, and as downloads when a VMware Cloud Director API client requests the transfer of content from VMware Cloud Director. A POST request initiates an upload, and a GET request initiates a download. The VMware Cloud Director transfer service facilitates uploads and downloads and provides temporary storage for files. Uploaded vApp templates and media images are made available as catalog items in the target catalog.

In addition to uploading, you can use the following operations to provision an organization with vApp templates, vApps, and media images:

Cloning

The VMware Cloud Director API `cloneVApp` operation creates a copy of a vApp in a specified VDC. You can specify whether to delete the source vApp after the operation completes. Deleting the source vApp after cloning it moves or renames it.

Capturing

The VMware Cloud Director API `captureVApp` operation creates a vApp template from a vApp and places the template in a specified catalog.

Importing

A system administrator can import a virtual machine from a vCenter server that is registered to the cloud. You can import the virtual machine as a vApp or as a vApp template. You can add an imported template to a catalog or download it as an OVF package.

Adopting

In the default configuration, the system automatically discovers vCenter VMs contained by in any resource pool that backs an organization VDC. A system administrator can also can configure an organization VDC to adopt existing vCenter resource pools and the VMs they contain. The system administrator can make these discovered VMs available for adoption by organization members.

Subscribing

Organizations that have the appropriate permissions can create catalogs with external subscriptions. These contents of these catalogs are downloaded from a catalog hosted on another instance of VMware Cloud Director, or any Web site that implements the VMware Content Subscription Protocol. See Create a Catalog With an External Subscription.

You can also create independent disks that are contained by an organization VDC and can be connected to any virtual machine in that VDC.

Read the following topics next:

- Upload an OVF Package to Create a vApp Template
- Download a vApp or vApp Template as OVF or OVA
- Upload a Media Image
- Download a Media Image
- Capturing and Importing vApps
- Discovering and Adopting vApps
- Managing Catalog Items
- Creating and Using Independent Disks
- View or Change the Owner of an Object
- Controlling Access to vApps and Catalogs

Upload an OVF Package to Create a vApp Template

A vCloud API client that has access to an OVF package can use a standard workflow to upload the package and create a vApp template.

The initial configuration of a vApp is established in the OVF package on which its source template is based. In the vCloud API, vApp templates are based OVF 1.0, an open standard format. For more information about OVF and how the vCloud API uses it, see About OVF.

An OVF package includes several kinds of files.

An OVF descriptor

An XML file that contains metadata that describe a virtual machine or collection of related virtual machines and the deployment environment they require. By convention, this file has the suffix `.ovf`.

Virtual disk files

The descriptor lists these files and includes information about their format.

An optional certificate

You can use this file to certify the authenticity of the package.

An optional manifest

Contains a SHA-1 digest of each of the files in the package.

Upload Workflow

The upload workflow for OVF packages uses a combination of vCloud API requests and standard HTTP file transfer requests.

1. The client makes a POST request to the catalog chosen to hold the template derived form the uploaded OVF. The request body specifies a name, description, and other parameters used when creating the template.
2. The server returns a `CatalogItem` that references a vApp template.
3. The client makes a GET request to the entity reference in the `CatalogItem` to retrieve the `VAppTemplate`, which includes an upload URL for the OVF descriptor. Because the template is not yet complete, the `VAppTemplate` element has `status="0"`.
4. The client makes a PUT request to the upload URL and supplies the OVF descriptor in the request body.
5. The server processes the descriptor and modifies the `vAppTemplate` to include an upload URL for each file listed in the `References` section of the descriptor. While the server is modifying the `vAppTemplate`, the client makes periodic requests for it and examines the response for additional upload URLs. When the response contains additional upload URLs that were not present in the initial response, template construction is complete.
6. The client uses PUT requests to upload each of the files.
7. If the OVF package includes a manifest file, the entire upload is validated against the contents of the manifest file.

Both monolithic and ranged, or chunked, PUT requests are supported. After starting an upload, a client can make periodic requests to assess its progress. After all of the files are uploaded (and validated if a manifest is present), the server processes them and updates the vApp template. When processing is complete, the server sets the value of the template's `status` attribute to `8`, indicating that it is ready for use. This status value indicates that all of the virtual machines in the template are powered off. For more information, including a complete list of possible status values and their meanings, see Object Creation Status.

Note If you have an OVF package that you want to deploy immediately as a vApp, without creating a vApp template and corresponding catalog item, make an `instantiateOvf` request. See Create a vApp From an OVF Package .

Restrictions on Uploaded Content

The VMware Cloud Director transfer service imposes the following restrictions on uploaded OVF content:

- You can upload either OVF 1.0 or OVF 1.1 content. OVF 1.1 packages are converted to OVF 1.0 for download, and any OVF 1.1 content is lost.
- Upload or download of packages that include OVF `ExtraConfig` elements typically require the user to have additional rights specific to the `ExtraConfig key` attribute. SeeSpecifying Advanced Virtual Machine Settings with ExtraConfig Elements.
- You cannot upload a compressed OVF package.
- If you upload an OVF package in which any `VirtualSystem` element has an `ovf:id` attribute value that is longer than 13 characters, the name of the `Vm` that represents that `VirtualSystem` in the `vAppTemplate` that the upload creates is rewritten as the first 13 characters of the `ovf:id` attribute followed by three digits. For example, `NewVirtualMachine1` and `NewVirtualMachine2` become `NewVirtualMac001` and `NewVirtualMac002`.

What to read next

Procedure

1 Initiating the OVF Upload

To initiate the OVF upload, a client makes a POST request to an `action/upload` link in the target catalog. The `type` of this link is `application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml`. The request body is an `UploadVAppTemplateParams` element.

2 Retrieving the Upload URL for the OVF Descriptor

After the vApp template and corresponding catalog item have been created, you must retrieve the template to get the upload URL for the OVF descriptor.

3 Uploading the OVF Descriptor

You upload the OVF descriptor by making a PUT request to the upload URL created for it in the `VAppTemplate`. The request body is the descriptor's `Envelope` element. If the request is valid, the server responds with a `200 OK` status.

4 Retrieving Additional Upload URLs

After an OVF descriptor is uploaded, the server validates it and, if it is valid, updates the corresponding template with upload URLs for each of the files referenced in the descriptor. You must retrieve the template to see these URLs.

5 Uploading Referenced Files

You can use a PUT request to upload each file that the vApp template references.

Initiating the OVF Upload

To initiate the OVF upload, a client makes a POST request to an `action/upload` link in the target catalog. The `type` of this link is `application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml`. The request body is an `UploadVAppTemplateParams` element.

The first step in uploading an OVF package is to request VMware Cloud Director to create a catalog item in the target catalog and a corresponding `vAppTemplate` object to represent the template that will be constructed from the upload.

Prerequisites

Verify that the following are true:

- You have an OVF package to upload.
- You are logged in as a user who has permission to upload OVF packages and create vApp templates.
- You know the URL of the target catalog that will receive the upload. Retrieve the XML representation of your organization to see a list of the catalogs that it contains.

Procedure

1 Find the `action/upload` link for vApp templates in the target catalog.

Retrieve the XML representation of the catalog using a request like the one shown in the request portion of Deployment Information in a VDC. The response contains an `action/upload` link, which has the following form:

```
<Link
   rel="add"
   type="application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml"
   href="https://vcloud.example.com/api/catalog/32/action/upload" />
```

2 Create an `UploadVAppTemplateParams` element that specifies the parameters for the vApp template that this request creates.

See the request portion of Initiating the Upload.

3 (Optional) If the OVF package includes a manifest, include a `manifestRequired="true"` attribute in the `UploadVAppTemplateParams` element.

Some OVF packages include a manifest document, which provides a checksum for each file in the package. When the `UploadVAppTemplateParams` element includes a `manifestRequired="true"` attribute, the set of `File` elements returned after you upload the OVF descriptor includes one for the manifest itself.

4 Make an HTTP POST request to the `upload` link that you retrieved in Step 1, supplying the `UploadVAppTemplateParams` element in the request body.

See the request portion of Initiating the Upload.

5 Examine the response.

The response, a `CatalogITem` element, contains an `Entity` element that contains a reference to the vApp template that will be constructed from the uploaded OVF. See the response portion of Initiating the Upload.

Results

The server creates a `VAppTemplate` object and a corresponding `CatalogItem` in the target catalog, and returns an XML representation of the `CatalogItem`. See the response portion of Initiating the Upload.

Example: Initiating the Upload

This example assumes an OVF package that has no manifest.

Request:

```
POST https://vcloud.example.com/api/catalog/32/action/upload
Content-Type: application/vnd.vmware.vcloud.uploadVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<UploadVAppTemplateParams
   name="Ubuntu Template"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Ubuntu vApp Template</Description>
</UploadVAppTemplateParams>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<CatalogItem
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Ubuntu Template"
   id="urn:vcloud:catalogitem:221"
   href="https://vcloud.example.com/api/catalogItem/221" ... >
  <Link
      rel="up"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/catalogItem/221/metadata" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.catalogItem+xml"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Description>Approved template for public FTP sites</Description>
   <Entity
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
      type="application/vnd.vmware.vcloud.vAppTemplate+xml"
      name="Ubuntu vApp Template"/>
</CatalogItem>
```

Retrieving the Upload URL for the OVF Descriptor

After the vApp template and corresponding catalog item have been created, you must retrieve the template to get the upload URL for the OVF descriptor.

Procedure

1 Examine the `CatalogItem` returned by the `upload` request to find the reference to the new vApp template.

The reference is the value of the `href` attribute of the `Entity` element, as shown here.

```
<Entity
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml"
   name="Ubuntu vApp Template" />
```

2 Retrieve the `VAppTemplate`.

See the request portion of OVF Descriptor Upload URL in a vAppTemplate.

3 Examine the template to find the upload URL for the OVF descriptor.

These URLs are contained in `Link` elements where `rel="upload:default"`.

Example: OVF Descriptor Upload URL in a vAppTemplate

This request uses the vApp template URL referenced in the `Entity` element shown in the response portion of Initiating the Upload.

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-111
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   ovfDescriptorUploaded="true"
   goldMaster="false"
   status="0"
   name="Ubuntu Template"
   id="urn:vcloud:vapptemplate:111"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml">
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"/>
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111" />
   <Description>Ubuntu vApp Template</Description>
   <Files>
      <File
         name="descriptor.ovf"
         bytesTransferred="0">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../descriptor.ovf" />
      </File>
   </Files>
   <Owner>
      ...
   </Owner>
   <Children />
   <LeaseSettingsSection>
      ...
   </LeaseSettingsSection>
   <CustomizationSection>
      ...
   </CustomizationSection>
</VAppTemplate>
```

The response body includes the following attributes:

- An ovfDescriptorUploaded attribute with a value of false, indicating that the OVF descriptor file is not uploaded.
- A status attribute with a value of 0, indicating that the file references in the descriptor are not uploaded. (A VAppTemplate with a status of 0 is said to be unresolved.)
- A goldMaster attribute, initially set to false.
- An id attribute. See Objects, References, and Representations.

The response body also includes a File element with an upload URL (rel="upload:default") for the OVF descriptor. The server creates the name attribute of this File element, which specifies a container that the server creates to receive the contents of the descriptor. The name attribute has no relation to the file name of the descriptor in the client's file system.

In addition to the File element, the response includes Owner, Children, LeaseSettingsSection, and CustomizationSection elements that the server creates and sets to their default contents. For more information about these elements, see the schema reference.

Uploading the OVF Descriptor

You upload the OVF descriptor by making a PUT request to the upload URL created for it in the `VAppTemplate`. The request body is the descriptor's `Envelope` element. If the request is valid, the server responds with a `200 OK` status.

Prerequisites

Verify that you have an upload URL for the OVF descriptor. See Retrieving the Upload URL for the OVF Descriptor.

Procedure

1 Upload the OVF descriptor.

Make a PUT request to the upload URL in the `VAppTemplate`. The upload URL for the OVF descriptor is in a `Link` element with the following form:

```
<Link
   rel="upload:default"
   href="https://vcloud.example.com/transfer/.../descriptor.ovf" />
```

Supply the OVF descriptor as the request body. The OVF descriptor contains a single `Envelope` element.

2 Verify that the request succeeded.

A response of the following form indicates that the request was valid and is being processed:

```
200 OK
```

Example: Uploading the OVF Descriptor

Request:

```
PUT https://vcloud.example.com/transfer/.../descriptor.ovf
Content-Type text/xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Envelope
   xmlns="http://schemas.dmtf.org/ovf/envelope/1"
   ... >
   ...
</Envelope>
```

Response:

```
200 OK
```

Retrieving Additional Upload URLs

After an OVF descriptor is uploaded, the server validates it and, if it is valid, updates the corresponding template with upload URLs for each of the files referenced in the descriptor. You must retrieve the template to see these URLs.

Procedure

1 Retrieve the `VAppTemplate` to verify that the OVF descriptor is uploaded.

   See the request portion of Upload URLs in a vAppTemplate.

2 Verify that the value of the template's `ovfDescriptorUploaded` attribute is `true`.

3 Examine the template to find the upload URLs for the files referenced in the OVF descriptor.

   These URLs are contained in `Link` elements where `rel="upload:default"`.

Example: Upload URLs in a vAppTemplate

This request uses the vApp template URL returned in Retrieving the Upload URL for the OVF Descriptor.

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-111
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   ovfDescriptorUploaded="true"
   goldMaster="false"
   status="0"
   name="Ubuntu Template"
   id="urn:vcloud:vapptemplate:111"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml">
   ...
   <Description>Ubuntu vApp Template</Description>
   <Files>
      <File
         size="3940"
         bytesTransferred="3940"
         name="descriptor.ovf">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../descriptor.ovf"/>
      </File>
      <File
         size="1024"
         bytesTransferred="0"
         name="manifest.mf">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../manifest.mf"/>
      </File>
      <File
         size="1950489088"
         bytesTransferred="0"
         name="disk0.vmdk">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../disk0.vmdk"/>
      </File>
   </Files>
   ...
</VAppTemplate>
```

In this example, which omits most of the additional elements shown in Initiating the Upload, the `ovfDescriptorUploaded` attribute has a value of `true` and the `status` attribute has a value of `0`. If the descriptor fails validation, `status` is set to `-1`, and the template contains a `Task` element whose `Error` element indicates the reason for the failure.

Each of the `File` elements includes an upload link where `rel="upload:default"` and several attributes.

size

The file size, taken from the `size` attribute of the `File` element in the OVF descriptor.

bytesTransferred

For all file references other than the descriptor, this attribute is initially set to a value of `0`, indicating that the upload has not begun. In the `File` element that references the OVF descriptor, the value of the `bytesTransferred` attribute is equal to the value of the `size` attribute, indicating that all the bytes in the descriptor were transferred.

name

The file name, taken from the `href` attribute of the `File` element in the OVF descriptor.

Note Upload URLs remain valid while a transfer session is in progress, and for a maximum of 60 minutes of transfer session idle time. A system administrator can change this default value. See Retrieve or Update System Settings.

Uploading Referenced Files

You can use a PUT request to upload each file that the vApp template references.

Prerequisites

- Verify that you uploaded the OVF descriptor. See Uploading the OVF Descriptor.

- Retrieve the upload URLs for all files in the package. See Retrieving Additional Upload URLs.

Procedure

1 Find the `upload:default` URL for the file you want to upload.

2 Use the `upload:default` URL to construct a PUT request for the file.

  The request specifies an upload URL and a content length in bytes. See Uploading File Data.

Results

After all the files are uploaded, the vApp template is complete, and has a `status` attribute value of `8`. If the upload included a manifest file, the server checks each file in the upload to verify that its checksum matches the one stated in the manifest. If a checksum does not match, the template's status attribute is set to `-1` and the template contains a `Task` element whose `Error` element indicates the reason for the failure.

Example: Uploading File Data

This example shows an upload request for one of the files that an OVF package requires. The upload request is a `Content-Length` header followed by the serialized file content.

Request:

```
PUT https://vcloud.example.com/transfer/.../disk0.vmdk
Content-length: 1950489088
   ...serialized contents of file disk0.vmdk...

EOF
```

Response:

```
200 OK
```

Monitoring the Progress of an Upload

After you initiate the upload of a file referenced by a vApp template, you can monitor the progress of the upload by periodically retrieving the vApp template and checking the value of the file's `bytesTransferred` attribute.

To monitor the progress of an upload, you can watch the `bytesTransferred` attribute of the file. Each `File` element in the template includes a `bytesTransferred` attribute whose value indicates the number of bytes that the server received.

Prerequisites

Verify that you initiated the upload of a file referenced by the vApp template.

Procedure

1 Make a GET request specifying the URL of the vApp template.

  See the request portion of Monitoring the Progress of an Upload.

2 Compare the values of the size and the bytesTransferred attributes of each File element.

  When these two values are equal, the file transfer is complete.

Results

After all the files are uploaded, the response includes final values for the bytesTransferred attribute of each File, and a Task that tracks the events leading up to resolution of the template with the uploaded files, as shown in Monitoring the Progress of an Upload.

Example: Monitoring the Progress of an Upload

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-111
```

The complete VAppTemplate body is returned. This example omits most of it for clarity.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   ...
   name="Ubuntu Template"
   id="urn:vcloud:vapptemplate:111"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml" ... >
   ...
   <Files>
      ...
      <File
         size="1950489088"
         bytesTransferred="500000000"
         name="disk0.vmdk">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../disk0.vmdk"/>
      </File>
   </Files>
   ...
</VAppTemplate>
```

Using Ranged PUT requests to Complete a Partial Upload

You typically need ranged PUT requests for very large uploads, especially when network bandwidth or latency might cause the operation to time out.

If the response to an upload progress request indicates that the upload terminated before it was complete, you can use the `size` and `bytesTransferred` values from the response to construct a ranged PUT request of the remaining contents, as shown in Ranged PUT Request to Complete a Partial Upload.

Procedure

1 Retrieve the `VAppTemplate` and find the `File` element that references the partially uploaded file.

2 Make a PUT request that specifies a `Content-Range` and `Content-Length` and includes the serialized contents of the range.

   For the `bytes` attribute of the `Content-Range`, specify the value of the `File` element's `bytesTransferred` attribute for the low end of the range and the value of its `size` attribute for the high end of the range. For `Content-Length`, subtract the value of the `File` element's `bytesTransferred` attribute from the value of its `size` attribute.

Example: Ranged PUT Request to Complete a Partial Upload

The following request completes the upload of the file `disk0.vmdk` shown in this fragment of a `VAppTemplate`.

```
<VAppTemplate
   ... >
   ...
   <Files>
      ...
      <File
         size="1950489088"
         bytesTransferred="500000000"
         name="disk0.vmdk">
         ...
      </File>
   </Files>
   ...
</VAppTemplate>
```

Request:

```
PUT https://vcloud.example.com/transfer/.../disk0.vmdk
Content-Range: bytes 500000000-1950489087/1950489088
Content-Length: 1450489088
   ...serialized contents of specified range...
EOF
```

Response:

```
200 OK
```

Download a vApp or vApp Template as OVF or OVA

You can download a vApp or vApp template object as an OVF package or OVA file. A vApp must powered off and undeployed before you can enable it for download.

An OVF package includes a descriptor and one or more virtual disk (VMDK) files. An OVA file is an archive that includes the descriptor and all other components. Downloads in either format are made available in two forms:

default

In the `default` format, the OVF descriptor for the vApp excludes identity information such as MAC addresses, BIOS UUIDs, and `NetworkConfigSection`. Ths information is typically specific to the VDC in which the vApp was last deployed, and is unlikely to be portable to other deployment environments.

identity

In the `identity` format, all deployment-specific information in the vApp or template is retained in the download.

When you enable a `VApp` or `VAppTemplate` object for download, the server adds download links in the following form to the XML representation of the object.

```
<Link type="text/xml"
      rel="download:default"
      href="https://vcloud.example.com/transfer/.../descriptor.ovf"/>
   <Link type="text/xml"
      rel="download:identity"
      href="https://vcloud.example.com/transfer/.../descriptor-with-id.ovf"/>
   <Link type="text/xml"
      rel="download:ovaDefault"
      href="https://vcloud.example.com/transfer/.../descriptor.ova"/>
   <Link type="text/xml"
      rel="download:ovaIdentity"
      href="https://vcloud.example.com/transfer/.../descriptor-with-id.ova"/>
```

The `rel` value of the link indicates the format of the download. When you make a download request to one of these links, the server constructs a download in the requested format and places it on the transfer server storage.

What to read next

Procedure

1 Enable a vApp or vApp Template for Download

Before you can download a vApp or vApp template, an administrator or privileged user must enable the object for download.

2 Download an OVF Descriptor or OVA File

To download the OVF descriptor or OVA file, make a GET request to the appropriate `download` link in the download-enabled `VApp` or `VAppTemplate` element.

3 Download a Referenced File

After you download the OVF descriptor of a vApp or vApp template, you can examine the contents of the descriptor to discover download URLs for `.vmdk` and other files in the package.

Enable a vApp or vApp Template for Download

Before you can download a vApp or vApp template, an administrator or privileged user must enable the object for download.

Prerequisites

- Verify that you are logged in as a user who has privileges to enable a vApp or vApp template for download.
- Verify that any vApp you plan to enable for download is powered off and undeployed. See Undeploy, Power Off, and Delete the vApp.

Procedure

1 Retrieve the XML representation of the `VApp` or `VAppTemplate` object.

2 Examine the representation to find its `action/enableDownload` link.

Every `VAppTemplate` element includes a link of the following form, where *id* is the `id` of the template:

```
<Link
   rel="enable"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-id/action/
enableDownload"/>
```

Every `vApp` element includes a similar link:

```
<Link
   rel="enable"
   href="https://vcloud.example.com/api/vApp/id/action/enableDownload"/>
```

3 Enable the object for download.

Make a POST request to the `action/enableDownload` URL, which you retrieved in Step 2. The response is a `Task` that tracks the completion of the enablement operation.

4 When the task completes, retrieve the representation of the object, which now contains OVF and OVA download URLs with `default` and `identity` contents.

Download URLs remain valid while a transfer session is in progress, and for a maximum of 60 minutes of transfer session idle time. A system administrator can change this default value. See Retrieve or Update System Settings.

Example: vApp Template with Download URLs

Request:

```
GET https://vcloud.example.com/api/vAppTemplate/vappTemplate-111
```

This download-enabled `VAppTemplate` object include four download `Link` elements. The `default` links exclude information such as MAC address, BIOS UUID, and `NetworkConfigSection` that is included in the `identity` download links.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   ovfDescriptorUploaded="true"
   status="8"
   name="Ubuntu Template"
   ... >
   ...
   <Link type="text/xml"
      rel="download:default"
      href="https://vcloud.example.com/transfer/.../descriptor.ovf"/>
   <Link type="text/xml"
      rel="download:identity"
      href="https://vcloud.example.com/transfer/.../descriptor-with-id.ovf"/>
   <Link type="text/xml"
      rel="download:ovaDefault"
      href="https://vcloud.example.com/transfer/.../descriptor.ova"/>
   <Link type="text/xml"
      rel="download:ovaIdentity"
      href="https://vcloud.example.com/transfer/.../descriptor-with-id.ova"/>
   ...
 ...
</VAppTemplate>
```

Download an OVF Descriptor or OVA File

To download the OVF descriptor or OVA file, make a GET request to the appropriate `download` link in the download-enabled `VApp` or `VAppTemplate` element.

When downloading an OVF or OVA, you can choose whether or not to include identity information such as lease settings and network connection details. Identity information is typically not portable to another deployment environment.

Prerequisites

- This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.
- Verify that you have a vApp or vApp template that is enabled for download. See Enable a vApp or vApp Template for Download.

Procedure

1. Retrieve the XML representation of the `VApp` or `VAppTemplate` object.
2. Examine the representation to find the download URLs for the OVF descriptor.

   The download URLs are contained in `Link` elements, each with a different value for the `rel` attribute.

Option	Description
Retrieve a descriptor that does not include identity information	Use the `download:default` or `download:ovaDefault`URL
Retrieve a descriptor that includes identity information.	Use the `download:identity` or `download:ovaIdentity` URL

3. Make a GET request to the URL that retrieves the download you want.

   If you are downloading an OVA file, you do not need to make any further requests. The server constructs the file and returns it in the response to the GET request. If you are downloading an OVF descriptor, you must take additional steps to download any other files that the descriptor references. See Downloading an OVF Descriptor and Download a Referenced File .

Example: Downloading an OVF Descriptor

This example downloads the OVF descriptor from the `download:default` URL shown in the `href` value of the `Link` shown in vApp Template with Download URLs. The response includes the entire `Envelope` element, only part of which appears here.

Request:

```
GET https://vcloud.example.com/transfer/..../descriptor.ovf
```

Response:

```
200 OK
Content-Type text/xml
...
<Envelope
   xmlns="http://schemas.dmtf.org/ovf/envelope/1"
   ... >
   ...
</Envelope>
```

Download a Referenced File

After you download the OVF descriptor of a vApp or vApp template, you can examine the contents of the descriptor to discover download URLs for `.vmdk` and other files in the package.

The OVF descriptor includes an `href` value for each file that the descriptor references. To retrieve one of these files, you must create a download URL for it by combining this `href` value with a URL derived from the download URL that you used to retrieve the descriptor. You must retrieve all of the referenced files to create a valid OVF package.

Prerequisites

- This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.
- Retrieve the OVF descriptor of a vApp or vApp template that has been enabled for download.

Procedure

1. For each `File` element in the `References` element of the descriptor, construct a download URL.
   a. Start with the URL that you used to download the descriptor.

      This URL is the `href` value of the `download:default` link that the template contains.

   b. Replace the final component of that URL with the value of the `href` attribute of the `File` element.
2. Use the constructed URLs to download each file.

   See Downloading a Referenced File.

Example: Downloading a Referenced File

The request URL shown in this example combines the URL used in the request portion of Downloading an OVF Descriptor with the file name shown in this `File` element:

```
<File
   ovf:href="disk0.vmdk"
   ovf:id="file1"
   ovf:size="1950489088"/>
```

Request:

```
GET https://vcloud.example.com/transfer/..../disk0.vmdk
```

Response:

```
200 OK
...
   ...serialized contents of file disk0.vmdk...

EOF
```

Note The downloaded package is valid only if the descriptor and all of its referenced files maintain the same relationship in the local file system that they had on the transfer server file system. In this case, the descriptor and `disk0.vmdk` were both in the same directory, which is the default arrangement.

Upload a Media Image

Uploading an ISO-format media image to a catalog creates a `Media` object and a corresponding `CatalogItem` object.

VMware Cloud Director supports using the vCloud API to upload media images to a catalog.

Note Media images in formats other than ISO can be uploaded, but are given an `imageType` of `other` in the catalog.

The workflow for uploading media images is similar to the one shown in Upload an OVF Package to Create a vApp Template .

Prerequisites

Verify that the following conditions are met:

- You have a media image to upload.
- You are logged in as a user who has permission to upload media images.
- You know the URL of the target catalog that will receive the upload. Retrieve the XML representation of your organization to see a list of the catalogs that it contains.

Procedure

1 Find the `add` link for `media` in the target catalog

   This link has the following form:

```
<Link
   rel="add"
   type="application/vnd.vmware.vcloud.media+xml"
   href="https://vcloud.example.com/api/catalog/32/action/upload" />
```

2 POST an `action/upload` request to the URL shown in Step 1

   The request body is a `Media` element. See the request portion of Upload a Media Image .

The server uses this information to create a `CatalogItem` and corresponding `Media` object, then returns the `CatalogItem` in its response. See the response portion of Upload a Media Image .

3 Use the URL in the `Entity` element of the `CatalogItem` to retrieve the `Media` object.

The Media element includes a `File` element that contains an `upload:default` URL.

4 PUT the media file contents to the `upload:default` link in the response.

The procedure is the same as the one shown in Uploading Referenced Files.

Example: Upload a Media Image

There are two steps to uploading a media file. The first step is to make an `action/upload` request to the catalog. The request body is a `Media` element that specifies the `size` of the ISO file and the name that you want to apply to the created `Media` object. The `imageType` attribute is optional, and must be set to a value of `iso` if you supply it.

Request:

```
POST https://vcloud.example.com/api/catalog/32/action/upload
Content-Type: application/vnd.vmware.vcloud.media+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Media
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="database.iso"
   size="51242131"
   imageType="iso">
   <Description>ISO database image</Description>
</Media>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<CatalogItem
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="database.iso"
   id="urn:vcloud:catalogitem:221"
   href="https://vcloud.example.com/api/catalogItem/221" ... >
  <Link
      rel="up"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/catalogItem/221/metadata" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.catalogItem+xml"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/catalogItem/221" />
   <Description>Approved template for public FTP sites</Description>
   <Entity
      type="application/vnd.vmware.vcloud.media+xml"
      name="database.iso"
      href="https://vcloud.example.com/api/media/254" />
</CatalogItem>
```

Examine the response to the `action/upload` request, then make a GET request to the URL in the `Entity` element of the `CatalogItem` to retrieve the `Media` object.

```
GET https://vcloud.example.com/api/media/254
```

The `Media` object includes an upload URL for the media file itself.

```

<Media ... >
   ...
   <Files>
      <File
         name="database.iso"
         bytesTransferred="0">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../database.iso" />
      </File>
   </Files>
   ...
</Media>
```

PUT the media file contents to the `upload:default` link in the response. The procedure is the same as the one shown in Uploading Referenced Files.

The upload URL remains valid while a transfer session is in progress, and for a maximum of 60 minutes of transfer session idle time. A system administrator can change this default value. See Retrieve or Update System Settings.

Download a Media Image

The VMware Cloud Director API supports downloading media images from a catalog.

Prerequisites

Verify that the following conditions are met:

- You are logged in as a user who has permission to download media images.
- You know the URL of the catalog item that references the media image.

Procedure

1. Retrieve the XML representation of the catalog and examine the catalog items that it contains.
2. Retrieve the catalog item that represents the media image.
3. Use the URL in the `Entity` element of the `CatalogItem` to retrieve the `Media` object.

   The `Media` element includes a `Link` element of the following form, where *id* is the `id` of the media image:

   ```
   <Link
      rel="enable"
      href="https://vcloud.example.com/api/media/id/action/enableDownload"/>
   ```
4. Enable the media image for download.

   Make a POST request to the `action/enableDownload` URL shown in Step 2. The response is a `Task` element.
5. When the task completes, retrieve the media item again.

   The `Media` object now includes a download URL for the media file.
6. Make a GET request to the `download:default` URL.

   The media file is downloaded to the current working directory.

Example: Download a Media Image

When you download a media file, you first enable the file for download.

Request:

```
POST https://vcloud.example.com/api/media/254/action/enableDownload
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
   ...
   operation="Enabling download of Media database.iso (254)" ... >
   ...
</Task>
```

The `Task` in the response tracks the creation of the downloadable image. When the task completes, retrieve the `Media` element again.

```
GET https://vcloud.example.com/api/media/254
```

The Media object now includes a download URL for the media file.

```
<Media ... >
   ...
   <Files>
      <File
         name="database.iso"
         bytesTransferred="0">
         <Link
            rel="download:default"
            href="https://vcloud.example.com/transfer/.../database.iso" />
      </File>
   </Files>
   ...
</Media>
```

The download URL remains valid while a transfer session is in progress, and for a maximum of 60 minutes of transfer session idle time. A system administrator can change this default value. See Retrieve or Update System Settings.

Capturing and Importing vApps

You can capture a vApp to create a vApp template from it. If you are a system administrator, you can also import vApps and vApp templates from vSphere.

As an administrator, catalog author, or vApp author, you can capture an undeployed vApp to create a vApp template in a catalog. Instantiating this template recreates the vApp from which the template was captured. Capturing a vApp in this way provides a way to save the results of composing, recomposing, or modifying a vApp after the vApp is undeployed. In addition, capturing a vApp preserves all vApp reconfiguration in template form. Although most elements of a vApp template are read-only, you can instantiate a template, modify the resulting vApp, and capture it to create a modified version of the template. See Capture a vApp as a Template

Importing vApps or vApp Templates from vSphere

A system administrator can import vApps and vApp templates from vSphere. See Import a Virtual Machine from vCenter .

Discovering and Adopting vApps

In the default configuration, an organization VDC discovers VMs that are created in any vCenter Server resource pool that backs the VDC. The system constructs a simplified vApp, owned by the system administrator, to contain each discovered virtual machine (VM). After the system administrator grants you access to a discovered vApp, you can reference the VM in it when you compose or recompose a vApp, or modify the vApp to adopt it and import it.

Discovered vApps contain exactly one VM, and are subject to several constraints that do not apply to vApps created in VMware Cloud Director. Whether or not you adopt them, they can be useful as a source of VMs to use when composing or recomposing a vApp.

Each discovered vApp is given a name that is derived from the name of the vCenter Server VM that it contains and a prefix specified by your organization administrator. When retrieved with a VMware Cloud Director API request, the `autoNature` element in a discovered vApp has a value of `true`. This value changes to `false` when the vApp is adopted.

If you want to discover additional vApps, a system administrator can use the VMware Cloud Director API to create organization VDCs that adopt specified resource pools available from a Provider VDC. vCenter Server VMs in these adopted resource pools appear in the new VDC as discovered vApps, and are candidates for adoption.

Note Virtual machines with IDE hard drives are discovered only if they are in powered off state.

Activating VM Discovery

Activating VM Discovery by Using the VMware Cloud Director API

VM discovery is active by default. To deactivate VM discovery for all organizations, a system administrator must update the value of the `VmDiscoveryEnabled` setting in the system's `GeneralSettings`. To deactivate VM discovery for all VDCs in an organization, an organization administrator must update the value of the `VmDiscoveryEnabled` setting in the `GeneralOrgSettings` for that organization. To deactivate VM discovery for an individual organization VDC, an organization administrator must update the value of the `VmDiscoveryEnabled` setting in the `AdminVdc` that represents the organization VDC.

To override the VM discovery default behavior, use the VMware Cloud Director API `/api/admin/extension/settings/general` to set the `AllowOverrideOfVmDiscoveryByOrgAndOVDC` parameter to `true`. When you set the parameter to `true`, you can modify the VM discovery settings at the organization and organization VDC level even if VM discovery is deactivated at the global level.

```
allow-override-of-vm-discovery-by-org-and-orgvdc = true
```

The `AllowOverrideOfVmDiscoveryByOrgAndOVDC` parameter is set to `null` by default and the global settings override all lower-level settings.

Using a VM from a Discovered vApp

After the system administrator grants you access to a discovered vApp, you can use its VM in the same ways you can use a VM that any other vApp or vApp template contains. For example, you can specify it when you compose or recompose a vApp. You can also clone a discovered vApp or modify its name, description, or lease settings without triggering the adoption process.

Adopting a Discovered vApp

You can adopt a discovered vApp by changing its vApp network or adding a VM to this vApp. After you adopted a discovered vApp, the system imports it and treats it as though it was created in VMware Cloud Director. When an adopted vApp is retrieved with a VMware Cloud Director API request, it includes an element named `autoNature`. This element has a value of `false` if the discovered vApp was adopted or was created in VMware Cloud Director. You cannot revert an adopted vApp to a discovered vApp.

If you delete or move the VM that a discovered vApp contains, the system also removes the containing vApp. This behavior does not apply to adopted vApps.

The vApp created to contain a discovered vCenter Server VM is similar to the one created when you manually import a VM as a vApp, but it is simplified in ways that might require you to modify it before you can deploy it in your VDC. For example, you might have to edit its networking and storage properties, and make other adjustments specific to the needs of your organization.

Note Adopting a virtual machine does not retain the VM reservation, limit, and shares settings that are configured in vCenter Server. Imported virtual machines obtain their resource allocation settings from the organization virtual data center on which they reside.

Managing Catalog Items

Catalog items are references to vApp templates and media files. If you have the appropriate rights, you can copy, move, rename, or delete catalog items in your organization's catalogs. You cannot modify catalog items in catalogs that have an external subscription.

After you add vApp templates or media files to a catalog, you might need to modify the `CatalogItem` objects that represent them. Your rights to manipulate catalog items depend on the source from which the catalog items were created.

- If you are a catalog author or an administrator, you can copy, move, delete, or rename catalog items that were uploaded, imported, or captured to a catalog that your organization owns, whether or not the catalog is published externally. Changes you make to an externally published catalog are replicated to all of the catalog's subscribers when those subscribers synchronize their copy of the catalog.
- You cannot make changes to catalog items in catalogs that have an external subscription.

Changes to a catalog item increment the version number of the item and its containing catalog. See Version Numbers.

In addition to providing storage for locally created vApp templates and media files, catalogs provide a flexible publication mechanism that supports distribution of content to other organizations and clouds. If your organization allows it, you can publish a catalog to external consumers. You can also subscribe to catalogs that external sources publish, although catalog items in such catalogs cannot be managed by subscribers. See Catalog Administration.

Copy or Move a Catalog Item

A `Catalog` object includes links that implement copy and move operations for the catalog items it contains.

To copy or move a catalog item from a source catalog to a target catalog, POST a `CopyOrMoveCatalogItemParams` element that contains a reference to the catalog item to move to the copy or move link of the target catalog.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that the target catalog does not have an external subscription.

Procedure

1 Retrieve the XML representation of the source catalog.

Use a request like this one:

```
GET https://vcloud.example.com/api/catalog/32
```

2 Examine the `Catalog` element to find the `CatalogItem` elements it contains.

Each `CatalogItem` in the `CatalogItems` container has `name`, `type`, and `href` attributes. If you need more information about a catalog item, you can retrieve it with a GET request to the URL in its `href` attribute.

3 Retrieve the XML representation of the target catalog.

4 Examine the `Catalog` element to find the `copy` and `move` links it contains.

These links have the following form:

```
<Link
   rel="copy"
   type="application/vnd.vmware.vcloud.copyOrMoveCatalogItemParams+xml"
   href="https://vcloud.example.com/api/catalog/44/action/copy" />
<Link
   rel="move"
   type="application/vnd.vmware.vcloud.copyOrMoveCatalogItemParams+xml"
   href="https://vcloud.example.com/api/catalog/44/action/move" />
```

5 Create a `CopyOrMoveCatalogItemParams` element that specifies the catalog item in the `Source` element.

See Copy a Catalog Item.

6 POST the CopyOrMoveCatalogItemParams to the appropriate link from the target catalog.

Option	Description
Copy the Catalog Item	POST the CopyOrMoveCatalogItemParams to the rel="copy" link.
Move the Catalog Item	POST the CopyOrMoveCatalogItemParams to the rel="move" link.

Example: Copy a Catalog Item

This request copies the catalog item shown in Retrieve a Catalog Item to another catalog. The response is a Task.

Request:

```
POST https://vcloud.example.com/api/catalog/44/action/copy
Content-Type: application/vnd.vmware.vcloud.copyOrMoveCatalogItemParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CopyOrMoveCatalogItemParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Ubuntu 10.04 Template">
   <Description>Reference copy of Ubuntu FTP Server</Description>
   <Source
      href="https://vcloud.example.com/api/catalogItem/221" />
</CopyOrMoveCatalogItemParams>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Copying Virtual Application Template Ubuntu 10.04 Template" ...>
...
</Task>
```

Change the Name or Description of a Catalog Item

Every CatalogItem object includes a rel="edit" link that you can use to modify the name or description of the catalog item.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that the target catalog does not have an external subscription.

Procedure

1 Retrieve the catalog item from the catalog.

2 Locate the rel="edit" link in the CatalogItem element.

3 Modify the retrieved `CatalogItem` element to change its name, description, or both.

   See Change the Name and Description of a Catalog Item.

4 Make a PUT request to the `href` value of the `rel="edit"` link in the `CatalogItem`, supplying the modified `CatalogItem` in the request body.

Example: Change the Name and Description of a Catalog Item

This request changes the name and the description of the catalog item shown in Retrieve a Catalog Item. The request body excludes components such as `Link` elements and `id` attributes that were present in the retrieved `CatalogItem`. These components are ignored if you include them in a request.

Request:

```
PUT https://vcloud.example.com/api/catalogItem/221
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CatalogItem
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="DEPRECATED Ubuntu Template">
   <Description>Deprecated. Use https://vcloud.example.com/api/vAppTemplate/vappTemplate-230
instead </Description>
   <Entity
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
      type="application/vnd.vmware.vcloud.vAppTemplate+xml"
      name="Ubuntu Template with vsftpd" />
</CatalogItem>
```

The response shows the modified `CatalogItem`.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.catalogItem+xml
...
<CatalogItem
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="DEPRECATED Ubuntu Template">
   <Description>Deprecated. Use https://vcloud.example.com/api/vAppTemplate/vappTemplate-230
instead </Description>
   <Entity
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"
      type="application/vnd.vmware.vcloud.vAppTemplate+xml"
      name="Ubuntu vApp Template" />
</CatalogItem>
```

Remove an Item from a Catalog

An organization administrator or a user with adequate permissions can remove a `CatalogItem` by making a DELETE request to its `rel="remove"` link.

Removing a `CatalogItem` also removes the referenced vApp template or media image from the catalog's storage.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that the target catalog does not have an external subscription.

Procedure

1. Retrieve the catalog item from the catalog.
2. Locate the `rel="remove"` link in the `CatalogItem` element.
3. Make a DELETE request to the `href` value of the `rel="remove"` link in the `CatalogItem`.

Example: Remove an Item from a Catalog

This request removes the source catalog item that was copied in Copy a Catalog Item.

Request:

```
DELETE https://vcloud.example.com/api/catalogItem/221
```

Response:

```
204 No Content
```

Synchronize a Catalog or Catalog Item

Catalogs that have external subscriptions are synchronized with their external sources by a background process that the system administrator controls. You can also force synchronization of individual catalog items or entire catalogs at any time.

Prerequisites

This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.

Procedure

1. Retrieve the XML representation of a catalog that has an external subscription.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/catalog/32
   ```

2. Examine the `Catalog` element to find the `CatalogItem` elements that it contains.

3 Examine the `Catalog` and `CatalogItem` element to find the `sync` links that they contain.

In catalogs, these links have the following form:

```
<Link
   rel="sync"
   href="https://vcloud.example.com/api/catalog/id/action/sync" />
```

In catalog items, these links have the following form:

```
<Link
   rel="sync"
   href="https://vcloud.example.com/api/catalogItem/id/action/sync" />
```

4 Synchronize the catalog or catalog item.

Make a POST request to the appropriate `action/sync` link.

Option	Description
Synchronize a Catalog	Make a POST request to the `action/sync` link in the `Catalog` element.
Synchronize a Catalog Item	Make a POST request to the `action/sync` link in the `CatalogItem` element.

Example: Synchronize a Catalog Item

This request synchronizes a single catalog item. The response is a task that tracks the progress of the synchronization.

Request:

```
POST https://vcloud.example.com/api/catalogItem/102/action/sync
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Synchronizing Catalog Item DB.iso (102)" ...>
...
</Task>
```

Creating and Using Independent Disks

Independent disks are stand-alone virtual disks that you create in organization VDCs. Administrators and users who have adequate rights can create, remove, and update independent disks, and connect them to virtual machines.

When you create an independent disk, it is associated with an organization VDC but not with a virtual machine. After the disk has been created in a VDC, the disk owner or an administrator can attach it to any virtual machine deployed in that VDC, detach it from a virtual machine, and remove it from the VDC.

Create an Independent Disk

To create an independent disk in an organization VDC, POST a `DiskCreateParams` element to the VDC's `disk` link.

To create an independent disk, you must specify its name and size. You can optionally include a description, and specify a storage profile to be used by the disk. After you have created the disk, you can modify its name, description, storage profile, and other propeties.

The owner of a disk is initially the user who created it. To change the owner, see View or Change the Owner of an Object.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1. Choose an organization VDC to contain the disk.
2. Create a `DiskCreateParams` element.

   You must specify the `size` (in bytes) and `name` of the independent disk. See the request portion of Create an Independent Disk.
3. POST the `DiskCreateParams` element you created in Step 2 to the URL for adding disks to the organization VDC.

   See the request portion of Create an Independent Disk.

Example: Create an Independent Disk

This example adds an independent disk to the organization VDC created in Add a VDC to an Organization . Because optional attributes `busType` and `busSubType` are omitted, a SCSI disk is created.

Request:

```
POST https://vcloud.example.com/api/vdc/44/disk
Content-Type: application/vnd.vmware.vcloud.diskCreateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<DiskCreateParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Disk
      name="500GB-SCSI"
      size="500000000000">
      <Description>500 GB SCSI Disk</Description>
   </Disk>
</DiskCreateParams>
```

The response, a subset of which appears here, is a `Disk` element that contains an embedded `Task` that tracks creation of the disk. Because the request did not specify a storage profile for the disk, it uses the default storage profile for the containing organization VDC. The response also includes `Link` elements that enable access to disk operations and metadata. While the disk is under construction, its `status` remains `0`.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.disk+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Disk
   xmlns="http://www.vmware.com/vcloud/v1.5"
   size="500000000000"
   status="0"
   name="500GB-SCSI"
   id="urn:vcloud:disk:128"
   type="application/vnd.vmware.vcloud.disk+xml"
   href="https://vcloud.example.com/api/disk/128"
   ... >
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/44" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/disk/128" />
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.disk+xml"
      href="https://vcloud.example.com/api/disk/128" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.owner+xml"
      href="https://vcloud.example.com/api/disk/128/owner" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.vms+xml"
      href="https://vcloud.example.com/api/disk/128/attachedVms" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/disk/128/metadata" />
   <Description>Independent Disk</Description>
   <Tasks>
      <Task
         ...
         operationName="vdcCreateDisk"
         ... >
         ...
      </Task>
   </Tasks>
   <StorageProfile
```

```
      type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
      name="bronze"
      href="https://vcloud.example.com/api/vdcStorageProfile/128" />
   <Owner
      type="application/vnd.vmware.vcloud.owner+xml">
      <User
         type="application/vnd.vmware.admin.user+xml"
         href="https://vcloud.example.com/api/admin/user/120" />
   </Owner>
</Disk>
```

Update an Independent Disk

You can update an independent disk to increase its capacity or change the storage profile that the disk uses.

Use this procedure to update an independent disk when it is not attached to a virtual machine.

When an independent disk is attached to a virtual machine, it is listed in the virtual machine's `VirtualHardwareSection` along with the other disks that are not independent of the virtual machine. You can update an attached independent disk by using the `reconfigureVm` operation. See Update Multiple Sections of a Virtual Machine.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve a reference to the independent disk.

  You can use a query like this one to retrieve references to all independent disks to which you have access:

  ```
  GET https://vcloud.example.com/api/query?type=disk&format=references
  ```

2 Retrieve the XML representation of the independent disk.

  Use one of the references returned by the query shown in Step 1, as shown in this example:

  Request:

  ```
  GET https://vcloud.example.com/api/disk/128
  ```

  The response is a `Disk` element.

Response:

```
<Disk
   ...
   name="500GB-SCSI"
   href="https://vcloud.example.com/api/disk/128">
   ...
</Disk>
```

3 Verify that the disk is not attached to a virtual machine.

Use the query service. A query like this one returns information about the returned in Step 2.

```
GET https://vcloud.example.com/api/query?type=adminDisk&filter=name==500GB-SCSI
```

The response includes an `isAttached` attribute. If this attribute has a value of `false`, the disk is not attached, and you can update it with a PUT request as shown in Update an Independent Disk. Otherwise, you can update it by using the `reconfigureVm` operation to update the appropriate `Item` in the `VirtualHardwareSection` of the virtual machine to which the disk is attached.

4 Modify the `Disk` element you retrieved in Step 2. See Update an Independent Disk.

Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

5 Update the `Disk` with your modifications.

a In the retrieved `Disk` element, find the `Link` element where `rel="edit"`.

b Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

Example: Update an Independent Disk

This example updates the independent disk created in Create an Independent Disk to change the storage profile.

Request:

```
PUT https://vcloud.example.com/api/disk/128
Content-Type: application/vnd.vmware.vcloud.disk+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Disk
   xmlns="http://www.vmware.com/vcloud/v1.5"
   size="500000000000"
   name="500GB-SCSI">
   <Description>Independent Disk</Description>
   <StorageProfile
      type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
      name="gold"
      href="https://vcloud.example.com/api/vdcStorageProfile/3" />
   <Owner type="application/vnd.vmware.vcloud.owner+xml">
      <User
         type="application/vnd.vmware.admin.user+xml"
         href="https://vcloud.example.com/api/admin/user/120" />
   </Owner>
</Disk>
```

The response is a `Task` element that tracks the update operation. When the task is complete, the object is updated.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Independent disk 500GB-SCSI (128)" ...>
...
</Task>
```

Remove an Independent Disk

To remove an independent disk, verify that no powered-on virtual machine is attached to it, then use a DELETE request to delete it.

A `Disk` element includes a link of the following form, which you can GET to return a list of virtual machines to which the disk is attached.

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.vms+xml"
   href="https://vcloud.example.com/api/disk/128/attachedVms" />
```

There are also two queries that you can use to return a list of virtual machines, the disks connected to them, and the VDC that contains them:

`vmDiskRelation`

Lists this information for `Vm` and `Disk` objects that you own.

`AdminvmDiskRelation`

Lists this information for all `Vm` and `Disk` objects in a cloud (system administrators only).

Note An independent disk can be attached to at most one virtual machine.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Verify that the independent disk is not connected to any virtual machine.

2 Delete the independent disk.

Make a DELETE request to the URL in the `rel="remove"` link in the `Disk`.

Results

The server starts a task to manage the events that lead up to the removal of the object, and returns a `Task` element that you can use to track the progress of the task.

Example: Remove an Independent Disk

Request:

```
DELETE https://vcloud.example.com/api/disk/128
```

Response:

```
202 Accepted
...
<Task
   ...
   operation="Deleting Disk (128)"
   ... >
</Task>
```

View or Change the Owner of an Object

You can view the owner of a `VApp`, `VAppTemplate`, `Disk`, or `Media` object by making a GET request to the object's `owner` link. If you have adequate rights, you can change the owner of a `Disk`, `VApp` object, or `VAppTemplate`, but not that of a `Media` object.

The initial owner of a `VApp`, `VAppTemplate`, `Catalog`, `Disk`, or `Media` object is the user who created it. Ownership is expressed in an `Owner` element that the object representation contains. This element includes a `User` element that references the owner. Object-specific rights to change ownership are included in several predefined roles. See #unique_79.

Starting with VMware Cloud Director API 31.0, you can change the owner of a `vAppTemplate` object.

Prerequisites

- To change the owner of a `Disk`, `VApp`, or `Catalog` object, you must be a **system administrator** or an **organization administrator**.
- To change the owner of a `vAppTemplate` object, you must be a **system administrator**, **organization administrator**, or **Catalog Author**.

Procedure

1 Retrieve the `Owner` element from the object.

This element includes a reference to the current owner and an `edit` URL you can use to change the owner. This request retrieves the owner of a vApp.

```
GET https://vcloud.example.com/api/vApp/vapp-7/owner
```

2 Modify the `Owner` element to specify a different `User`.

The user must be a member of the organization that contains the object.

Note You cannot modify the `Owner` of a `Media` object.

3 To change the owner, make a PUT request to the `Owner` element's `rel="edit"` URL and supply an `Owner` element in the request body.

The `User` element in the `Owner` element references the new owner. See Change the Owner of a vApp Template or vApp.

Example: Change the Owner of a vApp Template or vApp

- Request for changing the owner of a vApp template:

```
PUT https://vcloud.example.com/api/vAppTemplate/vappTemplate-44/owner
Content-type: application/vnd.vmware.vcloud.owner+xml;version=31.0
...
<?xml version="1.0" encoding="UTF-8"?>
<Owner
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <User
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/120" />
</Owner>
```

- Request for changing the owner of a vApp:

```
PUT https://vcloud.example.com/api/vApp/vapp-7/owner
Content-type: application/vnd.vmware.vcloud.owner+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Owner
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <User
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/120" />
</Owner>
```

Response:

```
204 No Content
```

Controlling Access to vApps and Catalogs

Upon creation, catalogs and vApps grant full access to their owners and no access to other users. TheVMware Cloud Director API access control mechanism enables object owners to retrieve or update these access controls as needed.

To retrieve or update the access controls on a vApp or catalog, use `controlAccess` links. The `controlAccess` links for catalogs are included when you retrieve the containing `AdminOrg`. The `controlAccess` links for a vApp are included in the `VApp` element itself.

VMware Cloud Director defines three levels of access:

`ReadOnly`

The `ReadOnly` access level grants rights to read or use the object.

`Change`

The `Change` access level includes all rights granted by `ReadOnly` access and grants additional rights to modify the object and its properties.

`FullControl`

The `FullControl` access level includes all rights granted by `Change` access and grants additional rights to change the owner of the object, share it, or delete it.

See Access Rights to VMware Cloud Director Objects for detailed information about the rights granted by each access level.

Access Control for vApps

An administrator or vApp owner can control access to a vApp.

Each `VApp` element includes two types of access control links:

- Links where `rel="down"`.

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.controlAccess+xml"
   href="https://vcloud.example.com/api/vApp/vapp-id/controlAccess/"/>
```

  Use this kind of link to retrieve the access control settings for the vApp identified in the `href` value.

- Links where `rel="controlAccess"`.

```
<Link
   rel="controlAccess"
   type="application/vnd.vmware.vcloud.controlAccess+xml"
   href="https://vcloud.example.com/api/vApp/vapp-id/action/controlAccess/"/>
```

Use this kind of link to specify new access control settings for the vApp identified in the `href` value. You specify the new access control settings in a `ControlAccessParams` element that you post to the URL that the `href` value of this link specifies. See Update vApp Access Controls for an example.

Access Control for Catalogs

An administrator can control access to a catalog. Each `Catalog` element includes two types of access control links:

- Links where `rel="down"`.

  ```
  <Link
     rel="down"
     type="application/vnd.vmware.vcloud.controlAccess+xml"
     href="https://vcloud.example.com/api/org/id/catalog/id/controlAccess/"/>
  ```

  Use this kind of link to retrieve the access control settings for the catalog identified in the `href` value.

- Links where `rel="controlAccess"`.

  ```
  <Link
     rel="controlAccess"
     type="application/vnd.vmware.vcloud.controlAccess+xml"
     href="https://vcloud.example.com/api/org/id/catalog/id/action/controlAccess/"/>
  ```

  Use this kind of link to specify new access control settings for the catalog identified in the `href` value. You specify the new access control settings in a `ControlAccessParams` element that you post to the URL that the `href` value of this link specifies.

Important These `controlAccess` links for catalogs are also returned in an `Org` element but their appearance in that context has been deprecated. They might be removed from `Org` elements in a future version of the VMware Cloud Director API .

Granting Access to All Members of an Organization

To specify access controls that apply to all members of an organization, an administrator can set `IsSharedToEveryone` to `true` and specify an access level in the `EveryoneAccessLevel` element.

The following `ControlAccessParams` element grants read access to all members of the organization.

```
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>true</IsSharedToEveryone>
   <EveryoneAccessLevel>ReadOnly</EveryoneAccessLevel>
</ControlAccessParams>
```

Granting Access to Individual Members of an Organization

To specify access controls that apply to specific users, an organization administrator can set `IsSharedToEveryone` to `false` and specify an access level in an `AccessSettings` element that the `ControlAccessParams` request contains.

An `AccessSettings` element is populated with one or more `AccessSetting` elements, each of which assigns an access level to the user identified in the `Subject` element. The following `ControlAccessParams` element grants full control to one user and read-only access to another user.

```
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/40"/>
         <AccessLevel>FullControl</AccessLevel>
      </AccessSetting>
      <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/45"/>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
   </AccessSettings>
</ControlAccessParams>
```

Viewing or Changing the Owner of a vApp or Catalog

Ownership of a `VApp` or `Catalog` object is expressed in an `Owner` element that you can retrieve from the object. This element contains a `User` element that identifies the owner with a reference to a specific user. The initial owner of an object is the user who created it.

A system administrator can view or change the owner of a `VApp` or `Catalog` object using the procedure documented in View or Change the Owner of an Object.

Access Rights to VMware Cloud Director Objects

Each access level supported by VMware Cloud Director grants one or more users a specific set of rights to an object.

VMware Cloud Director access levels are similar to roles in that they give a name to a set of rights. When you apply an access control to an object, you grant one or more users in your organization a set of rights to the object. Access rights are additive. You can make an object more accessible to users who have limited rights, but you cannot to restrict the rights that a user may already have. For example, an organization administrator retains full control of an object even if you apply `ReadOnly`access rights to it for all organization members.

Table 4-1. Access Levels and the Rights They Grant

	`FullControl`	`Change`	`ReadOnly`
Catalog: Add vApp from My Cloud	X	X	
Catalog: Change Owner	X		
Catalog: VCSP Publish Subscribe	X	X	
Catalog: Edit Properties	X	X	
Catalog: Publish	X	X	
Catalog: View Private and Shared Catalogs	X	X	X
Catalog: View Published Catalogs	X	X	X
vApp Template or Media: Copy	X	X	X
vApp Template or Media: Create or Upload	X	X	
vApp Template or Media: Edit	X	X	
vApp Template or Media: View	X	X	X
vApp Template: Checkout (Add to My Cloud)	X	X	X
vApp Template: Download	X	X	X
vApp: Change Owner	X		
vApp: Copy	X	X	X
vApp: Create or Reconfigure	X		
vApp: Delete	X		
vApp: Edit Properties	X	X	
vApp: Edit VM CPU	X	X	
vApp: Edit VM Hard Disk	X	X	
vApp: Edit VM Memory	X	X	
vApp: Edit VM Network	X	X	
vApp: Edit VM Properties	X	X	
vApp: Manage VM Password Settings	X		

Table 4-1. Access Levels and the Rights They Grant (continued)

	`FullControl`	`Change`	`ReadOnly`
vApp: Power Operations	X	X	
vApp: Sharing	X		
vApp: Use Console	X	X	X

Deploying and Operating vApps and Virtual Machines

5

A vApp object contains one or more virtual machines, and provides detailed specifications of those virtual machines and the networks to which they connect. The VMware Cloud Director API supports programmatic access to a range of self-service datacenter operations that allow users to create, configure, deploy, and operate vApps.

The initial configuration of a vApp and the virtual machines it contains is established in the OVF package on which its source template is based. In the VMware Cloud Director API, vApp templates are based on OVF 1.0. These templates can be retrieved from catalogs and transformed into virtual systems, called vApps, through a process called instantiation, which binds a template's abstract resource requirements to resources available in a VDC.

After a vApp has been created, using any of the methods described in About Instantiation, you can make further changes to its configuration using procedures like the ones shown in Chapter 6 Reconfiguring vApps and Virtual Machines . All configuration changes you make during instantiation or reconfiguration are discarded when the vApp is deleted, but you can preserve them by capturing the vApp as a template See Capture a vApp as a Template.

Starting with vCloud Director 9.5, vApps and virtual machines support IPv6 connectivity. You can assign IPv6 addresses to virtual machines connected to IPv6 networks.

About OVF

OVF is a widely accepted standard format that applies to many virtualization technologies.

- Virtual machines and appliances are distributed as OVF packages by many vendors.
- Many vendors, including VMware, offer tools that simplify creating and customizing OVF, support converting virtual machines on existing virtualization platforms to OVF, or both.
- OVF can express the complex relationships between virtual appliances in enterprise applications. The author of the appliance can handle most of the complexity, rather than the user who deploys it.
- OVF is extensible, allowing new policies and requirements to be inserted by ISVs and implemented by the virtualization platforms that support them without requiring changes to other clients, other platforms, or the VMware Cloud Director API itself.

Administrators and advanced users should become familiar with the details of the OVF standard before developing applications with the VMware Cloud Director API. The complete OVF specification document is available at http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf. An informative white paper on OVF is available at http://www.dmtf.org/standards/published_documents/DSP2017_1.0.0.pdf.

A virtual machine is typically made up of one or more virtual disk files that contain the operating system and applications that run on the virtual machine, and a configuration file containing metadata that describe how the virtual machine is configured and deployed. An OVF package includes these components, as well as optional certificate and manifest files. The package can be distributed and stored as a collection of individual files, or as a single archive (OVA) file. The VMware Cloud Director API supports upload and download of OVF files. It also supports download, but not upload, of OVA files.

About DMTF, CIM, and RASD

Virtual hardware in OVF packages elements is defined using an open standard framework established by the Distributed Management Task Force (DMTF). This framework, called the Common Information Model (CIM), defines virtual hardware resources using the ResourceAllocationSettingData (RASD) schema. In this schema, each class of virtual hardware is represented as an `Item` element with a specific `ResourceType`. Many VMware Cloud Director API operations that deploy and configure vApps and virtual machines require you to understand and sometimes modify RASD `Item` elements.

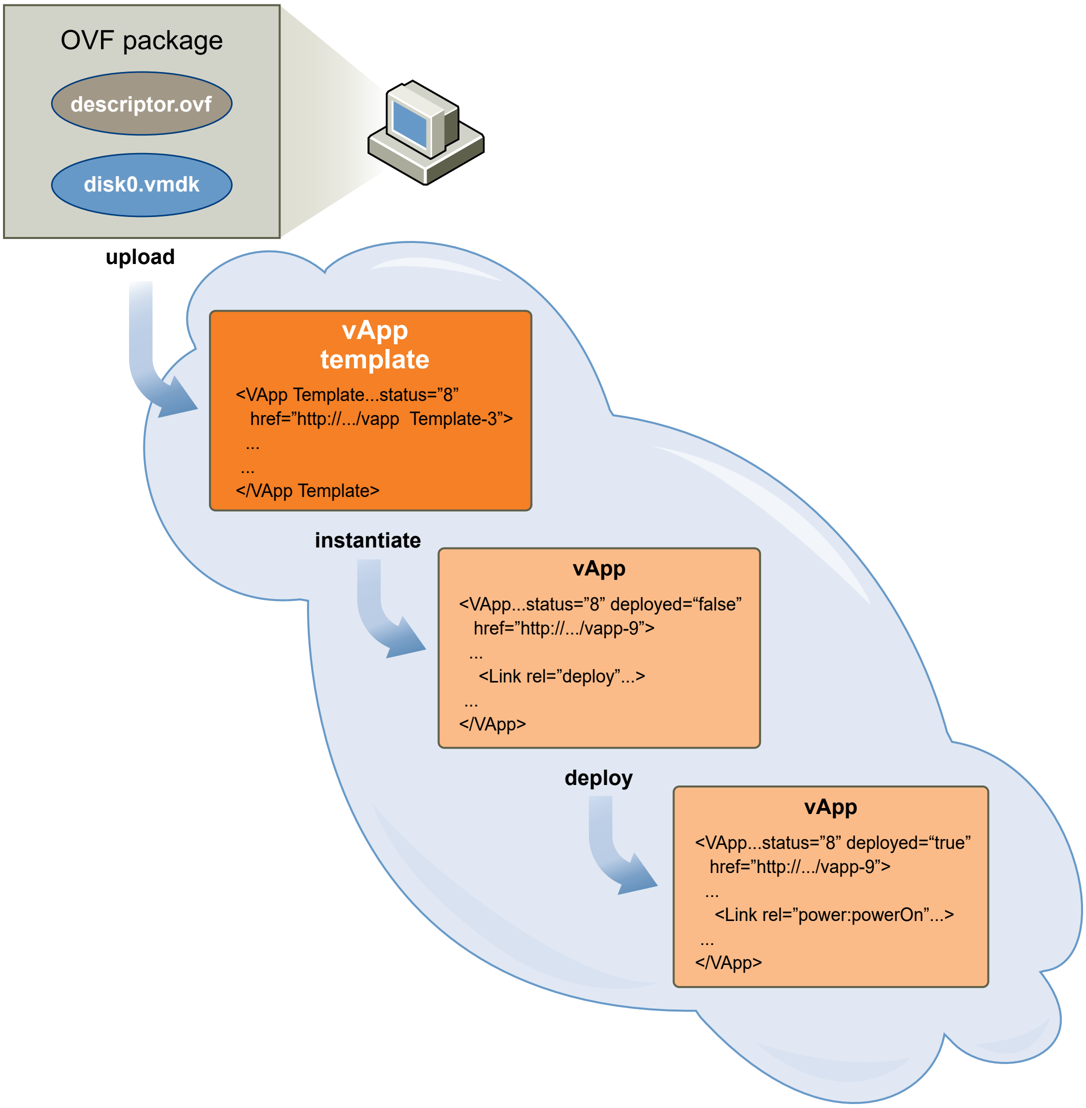
You can download the RASD schema files and related information from http://www.dmtf.org/standards/cim/cim_schema_v2191

vApp Life Cycle

A vApp contains one or more `Vm` elements, which represent individual virtual machines. It also contains information that defines operational details for the vApp and the virtual machines that it contains. The vApp lifecycle includes several distinct states:

- An OVF package, the form in which vApps are typically distributed.
- A vApp template, created when a client uploads an OVF package to a catalog.
- An undeployed vApp, created when a vApp template is instantiated without also being deployed, or a deployed vApp is undeployed.
- A deployed vApp, ready to be powered on and operated. Instantiation can include deployment, power-on, or both.

Figure 5-1. vApp State Transitions



```
<VApp Template...status="8"
   href="http://.../vapp  Template-3">
   ...
  ...
</VApp Template>
```

```
<VApp...status="8" deployed="false"
   href="http://.../vapp-9">
  ...
    <Link rel="deploy"...>
  ...
</VApp>
```

```
<VApp...status="8" deployed="true"
   href="http://.../vapp-9">
  ...
    <Link rel="power:powerOn"...>
  ...
</VApp>
```

For a full list of the VMware Cloud Director API user operations, see the *Cloud API Schema Reference* at https://code.vmware.com.

Read the following topics next:

- About Instantiation
- Create a vApp From a Template
- Modify Virtual Machine Hardware and Other Properties During vApp Template Instantiation
- Compose a vApp From Existing Virtual Machines
- Recompose a vApp to Add, Remove, or Reconfigure Virtual Machines
- Clone a vApp
- Create a vApp From an OVF Package
- Capture a vApp as a Template

- Update vApp Access Controls
- Create a VM-VM Affinity Rule
- Specifying Advanced Virtual Machine Settings with `ExtraConfig` Elements
- Operate a vApp
- Create a Virtual Machine with Custom Resource Allocation Settings

About Instantiation

Instantiation binds the abstract requirements for resources such as memory, CPU, and networking expressed in a vApp, VM, or vApp template to concrete instances of appropriate resources in a target VDC.

vApp templates and the vApps and virtual machines created from them include detailed specifications of virtual hardware, network requirements, and other properties like computer names and descriptions, guest operating system configurations, storage leases, end user license agreement (EULA) text, and so on. Any time you create a vApp from a template or include a vApp or virtual machine in a composed vApp, you have the opportunity to modify those specifications so that the resulting configuration meets the needs of the workload. Instantiation operations use a POST request and create a vApp with a configuration you specify in the request body.

Not all configuration details of a vApp or virtual machine can be modified during instantiation. As an adjunct to instantiation, you can use various reconfiguration operations to update an existing vApp or virtual machine. See Chapter 6 Reconfiguring vApps and Virtual Machines .

Instantiation Parameters

The `InstantiationParams` element is a generic parameter-passing mechanism that can apply to a vApp or virtual machine.

Table 5-1. Requests That Allow `InstantiationParams`

Request	Request Body	Description
`instantiateVAppTemplate`	`InstantiateVAppTemplateParams`	Creates a vApp from a vApp template. This simple form of instantiation is limited to creating a vApp that includes the set of virtual machines defined in the template.
`composeVApp`	`ComposeVAppParams`	Creates a vApp composed from any combination of vApp templates and virtual machines. Virtual machines referenced in the request body must be powered off and cannot have an independent disk attached. They can be sourced from any vApp or vApp template accessible to you. Any vApp referenced in the request body contributes all of its virtual machines to the composed vApp, but its vApp-level configuration details (such as vApp networks and lease settings) are ignored and replaced by the vApp-level instantiation parameters supplied in the request body.
`recomposeVApp`	`RecomposeVAppParams`	Edits a vApp to add, remove, or reconfigure virtual machines. Virtual machines referenced in the request body must be powered off and cannot have an independent disk attached. They can be sourced from any vApp or vApp template accessible to you. Any vApp added contributes all of its virtual machines to the composed vApp, but its vApp-level configuration details (such as vApp networks and lease settings) are ignored and replaced by the vApp-level instantiation parameters supplied in the request body.
`cloneVApp`	`CloneVAppParams`	Creates a copy of an existing vApp. You can include vApp-level instantiation parameters that apply to the copy. You can also include `SourcedItem` elements that apply instantiation parameters to virtual machines in the vApp.
`instantiateOvf`	`InstantiateOvfParams`	Creates a vApp or virtual machine from an OVF upload. This simple form of instantiation is limited to creating the vApp or virtual machine defined in the uploaded OVF package.

The set of elements and attributes that are allowed within an `InstantiationParams` element depends on whether the `InstantiationParams` apply to a vApp template or virtual machine.

Instantiation Parameters for vApps

An `instantiateVAppTemplate`, `composeVApp`, or `recomposeVApp` request can modify certain properties of a vApp by including an `InstantiationParams` element at the vApp level. The placement of vApp-level `InstantiationParams` depends on the type of request you are making:

- In an `instantiateVAppTemplate` request, vApp-level instantiation parameters are included at the root level of the `InstantiateVAppTemplateParams` request body, preceding the `Source` element that references the template you are instantiating. See Instantiation Parameters for vApps
- In a `composeVApp` request, vApp-level instantiation parameters are included in the `SourcedItem` whose `Source` element references the vApp template being instantiated or included in the composition.
- In a `recomposeVApp` request, which specifies a vApp in the request URL, the vApp-level instantiation parameters are included at the root level of the `RecomposeVAppParams` request body.
- In an `instantiateOvf` request, vApp-level instantiation parameters are included at the root level of the `InstantiateOvfParams` request body.

`InstantiationParams` for a vApp can include any of the following elements:

`LeaseSettingsSection`

Defines the terms of storage and deployment leases for the vApp. If this section is omitted, the vApp inherits the default lease settings of the containing organization.

`NetworkConfigSection`

Defines the properties of the vApp network and specifies how it is connected to one or more organization VDC networks. Unless you intend to create a vApp that has no connection to any network, you must include this section in your `InstantiationParams`.

`StartupSection`

Defines the order in which the virtual machines in the vApp start up and shut down. If this section is omitted, the startup and shutdown order of virtual machines in the vApp is indeterminate.

Instantiation Parameters for Virtual Machines

An `instantiateVAppTemplate`, `composeVApp`, or `recomposeVApp` request can modify certain properties of individual virtual machines by including `InstantiationParams` in the `SourcedItem` whose `Source` element references the virtual machine.

`InstantiationParams` for a virtual machine can include any of the following elements:

`VirtualHardwareSection`

Contains a description of the virtual hardware supported by a virtual machine. Each hardware resource is defined in an `Item` element. Instantiation parameters for a virtual machine can include individual items of the following types:

- CPU (RASD resource type 3)
- Memory (RASD resource type 4)
- Hard disk (RASD resource type 17)

Important Changes to most `Item` elements in a `VirtualHardwareSection` are ignored by the `composeVApp` operation.

`GuestCustomizationSection`

Contains guest customization parameters for the virtual machine.

`OperatingSystemSection`

Specifies the guest operating system installed on the virtual machine.

`ProductSection`

Contains information about software installed on the virtual machine.

`NetworkConnectionSection`

Specifies how the virtual NIC devices on the virtual machine are connected to the vApp network.

`VmCapabilities`

Allows you to specify virtual machine capabilities such as hot-add of memory or CPU.

About vApp Networks

Every vApp contains a vApp network. Virtual machines in the vApp connect to this network, which can be isolated from other networks or connected to an organization VDC network.

A vApp network is a logical network that controls how the virtual machines in a vApp connect to each other and to organization VDC networks. You create a vApp network when you make an `instantiateVAppTemplate`, `composeVApp`, `recomposeVApp`, or `instantiateOvf` request. The network is created when the vApp is deployed, and deleted when the vApp is undeployed. All nonisolated virtual machines in the vApp connect to a vApp network, as specified in their `NetworkConnectionSection` elements.

Every VApp element includes a link that you can use to retrieve details of its vApp network.

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.vAppNetwork+xml"
   name="isoNet1"
   href="https://vcloud.example.com/api/network/94 />
```

A GET request to this link returns a read-only VAppNetwork element with the configuration specified in the InstantiationParams used when the vApp was created or composed. To modify an existing vApp network, retrieve its NetworkConfigSection and use the edit link it contains, as shown in Update a vApp Network Configuration.

vApp Network Configurations

The configuration of a vApp network, represented by a NetworkConfig element contained in the NetworkConfigSection of your InstantiationParams, includes the following information

- A name for the network, specified in the networkName attribute of the NetworkConfig element. The instantiation parameters must create a vApp network whose name matches the value of the network attribute of the NetworkConnection of each Vm element in the template. If this attribute has the value none or is missing, the Vm can connect to any network. If the template contains Vm elements that specify different names for their network connections, you must create a vApp network for each.

  Note When you create a vApp network where the FenceMode is bridged, the networkName of the vApp network must match the name of the ParentNetwork. This requirement is enforced by the composeVapp operation. The instantiateVappTemplate operation automatically corrects a name mismatch by changing the value of the network attribute in the NetworkConnection element of the VApp.

- A Configuration element that specifies network configuration details.
  - For routed and directly connected networks, the ParentNetwork element contains a reference to the organization VDC network that the vApp network connects to. The FenceMode element controls how the two networks connect. Specify a FenceMode of bridged for a direct connection to the parent network, or natRouted to specify a routed connection controlled by network Features such as a NatService or FirewallService. If you want the organization network to be isolated, with no external connection, omit the ParentNetwork element and specify the FenceMode as isolated.
  - The Features element defines network services, such as DHCP, firewall, network address translation, and static routing, provided to virtual machines in the vApp.
  - Additional modifiable elements like IpScopes and RetainNetInfoAcrossDeployments, and read-only elements such as SyslogServerSettings and RouterInfo. For more information about the type and scope of these elements, see the schema reference.

- Network pool resources required by an `isolated` or `natRouted` vApp network are allocated by the system from the pool associated with the VDC in which the vApp is deployed.

Network Services in vApp Networks

The `Features` element of a vApp `NetworkConfigSection` defines the network services available to virtual machines in the vApp.

A vApp network can be configured to provide many of the same kinds of services available in an organization VDC network. Configuration parameters for these services are similar to those of their counterparts on an Edge Gateway, but scoped to the needs of a vApp network.

For more information about VMware Cloud Director networks, see About VMware Cloud Director Networks. For more information about network services for organization VDC networks, see Configure Edge Gateway Services

DHCP Service

A `DhcpService` element defines an IP address range and lease policies for a DHCP service that can be used by virtual machines in the vApp. Unlike a DHCP service in an Edge Gateway, it can support only a single IP address range, as shown in this example.

```
<DhcpService>
   <IsEnabled>true</IsEnabled>
   <DefaultLeaseTime>3600</DefaultLeaseTime>
   <MaxLeaseTime>7200</MaxLeaseTime>
   <IpRange>
      <StartAddress>192.168.3.2</StartAddress>
      <EndAddress>192.168.3.99</EndAddress>
   </IpRange>
</DhcpService>
```

Firewall Service

A `FirewallService` element defines firewall rules that, when matched, block or allow incoming or outgoing traffic on the vApp network. A firewall rule in a vApp network can specify the destination as a combination of address and port, or as a specific virtual NIC in a `Vm`. This `FirewallService` allows TCP traffic to ports 21 and 22.

```
<FirewallService>
   <IsEnabled>true</IsEnabled>
   <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>FTP Rule</Description>
      <Policy>allow</Policy>
      <Protocols>
         <Tcp>true</Tcp>
      </Protocols>
      <DestinationPortRange>21</DestinationPortRange>
      <DestinationIp>10.147.115.1</DestinationIp>
      <SourcePortRange>any</SourcePortRange>
      <SourceIp>any</SourceIp>
      <EnableLogging>false</EnableLogging>
   </FirewallRule>
   <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>SSH Rule</Description>
      <Policy>allow</Policy>
      <Protocols>
         <Tcp>true</Tcp>
      </Protocols>
      <DestinationPortRange>22</DestinationPortRange>
      <DestinationIp>10.147.115.1</DestinationIp>
      <SourcePortRange>any</SourcePortRange>
      <SourceIp>any</SourceIp>
      <EnableLogging>false</EnableLogging>
   </FirewallRule>
</FirewallService>
```

You can see this example in the context of a vApp `NetworkConfigSection` in Update a NetworkConfigSection

An alternate implementation of the second `FirewallRule` in this example includes a `DestinationVm` element that specifies the destination as a specific virtual NIC (identified in the `VmNicId` element) in a specific `Vm` (identified in the `VAppScopedVmId` element. The value of `VAppScopedVmId` is taken from the `VAppScopedLocalId` element of the `Vm` and the `VmNicId` value is taken from its `PrimaryNetworkConnectionIndex`. See Configuration Links in a Vm Element. The `IpType` is set to `assigned`, indicating that the NIC retains its assigned IP address. If you set `IpType` is set to `NAT`, the IP address of the NIC is its translated address.

```
<FirewallRule>
   <IsEnabled>true</IsEnabled>
   <Description>allow ssh to a specific NIC in a specific Vm</Description>
   <Policy>allow</Policy>
   <Protocols>
      <Tcp>true</Tcp>
   </Protocols>
   <DestinationPortRange>22</DestinationPortRange>
   <DestinationVm>
      <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e8e5b</VAppScopedVmId>
      <VmNicId>0</VmNicId>
      <IpType>assigned</IpType>
   </DestinationVm>
   <SourcePortRange>Any</SourcePortRange>
   <SourceIp>Any</SourceIp>
   <EnableLogging>false</EnableLogging>
</FirewallRule>
```

NAT Service

A NatService element defines network address translation services to virtual machines on the network. This simple NatService defines a single rule that implements an IP translation NAT strategy for a single Vm.

```
<NatService>
   <IsEnabled>true</IsEnabled>
   <NatType>ipTranslation</NatType>
   <Policy>allowTraffic</Policy>
   <NatRule>
      <OneToOneVmRule>
         <MappingMode>automatic</MappingMode>
         <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e8e5b</VAppScopedVmId>
         <VmNicId>0</VmNicId>
      </OneToOneVmRule>
   </NatRule>
</NatService>
```

You can see this example in the context of a vApp NetworkConfigSection in Update a NetworkConfigSection

A NatService element like this one configures the service to use port forwarding instead of IP translation. Instead of using a OneToOneVmRule, which specifies one external IP address to one NIC, it uses a VmRule element, which enables port forwarding by allowing one external IP address to be forward to different ports on different virtual machines.

```
<NatService>
   <IsEnabled>true</IsEnabled>
   <NatType>portForwarding</NatType>
   <Policy>allowTraffic</Policy>
   <NatRule>
      <VmRule>
         <ExternalPort>22</ExternalPort>
         <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e8e5b</VAppScopedVmId>
         <VmNicId>0</VmNicId>
         <InternalPort>22</InternalPort>
         <Protocol>TCP</Protocol>
      </VmRule>
   </NatRule>
</NatService>
```

Static Routing Service

A StaticRoutingService specifies static routes to other networks. In addition to creating static routes from organization VDC networks on an EdgeGateway (see Static Routes Between Organization VDC Networks, you can create static routes between vApp networks if they both define the same ParentNetwork. Assume two vApp networks that have the following properties:

- The Configuration of the vApp network in vApp1 has a RouterInfo element whose ExternalIp value is 192.168.0.100.

- The Configuration of the vApp network in vApp2 has a RouterInfo element whose ExternalIp value is 192.168.0.101.
- Both vApp networks have the same ParentNetwork, an organization VDC network whose network specification in CIDR notation is 192.168.0.0/24.

You can enable static routing between these two vApp networks by inserting a StaticRoutingService element in the Features of each vApp network Configuration. This excerpt from the NetworkConfigSection of vApp1 shows the network's Configuration and Features elements..

```
<!-- Static route from vApp network 1 to vApp network 2 -->
<Configuration>
   ...
   <Features>
      <StaticRoutingService>
         <IsEnabled>true</IsEnabled>
         <StaticRoute>
            <Name>TovAppNet2</Name>
            <Network>192.168.2.0/24</Network>
            <NextHopIp>192.168.0.101</NextHopIp>
            <Interface>External</Interface>
         </StaticRoute>
       </StaticRoutingService>
   </Features>
   ...
</Configuration>
```

This is a similar excerpt from the NetworkConfigSection of vApp2.

```
<!-- Static route from vApp network 2 to vApp network 1 -->
<Configuration>
   ...
   <Features>
      <StaticRoutingService>
         <IsEnabled>true</IsEnabled>
         <StaticRoute>
            <Name>TovAppNet1</Name>
            <Network>192.168.1.0/24</Network>
            <NextHopIp>192.168.0.100</NextHopIp>
            <Interface>External</Interface>
         </StaticRoute>
      </StaticRoutingService>
   </Features>
   ...
</Configuration>
```

Create a vApp From a Template

An instantiateVAppTemplate request creates a vApp from a vApp template.

To create a vApp from a vApp template, you must bind the template's abstract resource requirements, such as network connections, storage resources, memory, and CPU capacity, to appropriate resources in the target VDC. This binding operation is called instantiation.

For an example of a simple instantiation request, see Deploy the vApp. You can also specify additional parameters as part of instantiation.

Template contents that might influence composition of the request body include the following elements in the vApp itself:

- A NetworkConfigSection that defines a vApp network to which virtual machines in this vApp can connect.
- One or more EulaSection elements that specify licensing terms or other conditions that you must accept before creating the vApp. The InstantiateVAppTemplateParams element can include an AllEULAsAccepted element whose value indicates whether you accept all EULA terms included in the template. If a vApp template includes any ovf:EulaSection elements, AllEULAsAccepted must be set to a value of true. Otherwise, instantiation fails.
- A LeaseSettingsSection. If this section is present and specifies settings that are appropriate for the vApp, you do not need to modify it. If it is absent or empty, the vApp is created with your organization’s default lease settings. If you specify new lease settings in a LeaseSettingsSection that you provide as part of instantiation, those settings replace any existing settings and override your organization's defaults.

Template contents that might influence composition of the request body include the following elements in the virtual machines that the template contains.

- A NetworkConnectionSection that specifies network connection details for a virtual machine. Unless you want to create a vApp in which none of the virtual machines are connected to a network, your instantiation parameters must include at least one NetworkConfigSection that defines a vApp network, and that section must include a NetworkConfig element whose networkName attribute value matches the value of the network attribute of the NetworkConnection of each Vm in the template. If this attribute has the value none or is missing, the Vm can connect to any network. If the template contains Vm elements that specify different names for their network connections, you must create a vApp network for each.

Prerequisites

- This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.
- Review the current configuration of the vApp and its virtual machines. Configuration parameters in the VAppTemplate such as its NetworkConfigSection and LeaseSettingsSection affect all virtual machines in the vApp. Configuration parameters for individual virtual machines are defined in Vm elements in the VAppTemplate. A virtual machine's network connections are defined in its NetworkConnectionSection, and its hardware configuration is

defined in its VirtualHardwareSection. To view the configuration of a virtual machine in a vApp template, you can retrieve the template and examine the VirtualHardwareSection of that Vm, or you can download the OVF descriptor of the vApp template, as shown in Download an OVF Descriptor or OVA File .

Procedure

1. Retrieve the XML representation of the vApp template.

   Make a GET request to the URL provided in the href attribute of the Entity contained by the CatalogItem that references the template. You can also use the query service to return a list of references to vApp templates that you can access.

2. Examine the template to determine the set of instantiation parameters that the request must include.

3. Create an InstantiateVAppTemplateParams element.

   See Instantiate a vApp Template and Modify Virtual Machine Name, Description, and Storage Profile for guidelines.

4. Make a POST request to the action/instantiateVAppTemplate URL of the VDC.

   Supply the InstantiateVAppTemplateParams element as the request body.

Results

The server takes the requested action and returns a VApp element. The element has a status attribute value of 0, meaning it is unresolved because the vApp is still being constructed. It also contains a Task element that tracks the progress of the request.

Example: Instantiate a vApp Template and Modify Virtual Machine Name, Description, and Storage Profile

An InstantiateVAppTemplateParams request body includes a root-level InstantiationParams element that provides instantiation parameters for the vApp. To modify properties of any virtual machine in the template during instantiation, include a SourcedItem element that references the virtual machine and provides InstantiationParams for it. Virtual machines referenced from SourcedItem elements in an InstantiateVAppTemplateParams must be members of the Children collection of the vApp template being instantiated.

```
<!-- InstantiateVAppTemplateParams request body prototype -->
<InstantiateVAppTemplateParams>
   <InstantiationParams>
      <!-- vapp-level params -->
   </InstantiationParams>
   <Source href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"/>
   <SourcedItem>
      <Source href="A Vm in template-111">
         <InstantiationParams>
            <!-- Vm-level params -->
         </InstantiationParams>
      </Source>
   </SourcedItem>
</InstantiateVAppTemplateParams>
```

This `InstantiateVAppTemplateParams` request extends the request shown in Deploying a vApp to include additional elements in its `InstantiationParams`:

- A `LeaseSettingsSection` that specifies custom lease settings, overriding the settings that would otherwise be inherited from the organization.
- An acknowledgement of `EulaSection` acceptance, supplied in the `AllEULAsAccepted` element. If the template does not include `EulaSection` elements, you can omit this acknowledgement.
- A `SourcedItem` element whose `Source` element references a virtual machine in the template. In this example, the `SourcedItem` contains:
  - a `VmGeneralParams` element that specifies a new `name` and `Description` for a virtual machine in the template. If you omit this element, instantiation creates the virtual machine with the `name` and `Description` specified in the template.
  - a `StorageProfile` element that specifies a storage profile to be used for this virtual machine. If you omit this element, the virtual machine uses the default storage profile defined by the containing VDC.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/instantiateVAppTemplate
Content-Type: application/vnd.vmware.vcloud.instantiateVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Linux FTP server"
   deploy="true"
   powerOn="true"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Example FTP Server</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks</ovf:Info>
         <NetworkConfig networkName="vAppNetwork">
            <Configuration>
               <ParentNetwork href="https://vcloud.example.com/api/network/54"/>
               <FenceMode>bridged</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
      <LeaseSettingsSection
         type="application/vnd.vmware.vcloud.leaseSettingsSection+xml">
         <ovf:Info>Lease Settings</ovf:Info>
         <StorageLeaseInSeconds>172800</StorageLeaseInSeconds>
         <StorageLeaseExpiration>2014-04-25T08:08:16.438-07:00</StorageLeaseExpiration>
      </LeaseSettingsSection>
   </InstantiationParams>
   <Source href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"/>
   <SourcedItem>
      <Source href="https://vcloud.example.com/api/vAppTemplate/vm-4"/>
      <VmGeneralParams>
         <Name>ftp1</Name>
         <Description>Primary FTP Server Instance</Description>
         <NeedsCustomization>true</NeedsCustomization>
      </VmGeneralParams>
      <StorageProfile href="https://vcloud.example.com/api/vdcStorageProfile/33">
      </StorageProfile>
   </SourcedItem>
   <AllEULAsAccepted>true</AllEULAsAccepted>
</InstantiateVAppTemplateParams>
```

The response is a sparsely populated `VApp` element, as shown in the response portion of Deploying a vApp.

Modify Virtual Machine Hardware and Other Properties During vApp Template Instantiation

Instantiation parameters for a vApp template can modify the virtual hardware configuration, including network connections, of the virtual machines defined in the template.

An `InstantiateVAppTemplateParams` request body that incorporates one or more `SourcedItem` elements supports a number of changes to the configuration, including the virtual hardware configuration, of virtual machines in the template. For example:

- Change the `name`, `Description`, and `NeedsCustomization` properties of the virtual machine.
- Specify a storage profile for the virtual machine.
- Specify a storage profile for any of the virtual machine's hard disks, overriding the virtual machine's default storage profile.
- Specify how the NICs in the virtual machine connect to vApp networks defined in the `NetworkConfigSection` of the vApp.
- Specify virtual machine capabilities.
- Increase the capacity of the virtual machine's SATA or SCSI disks.
- Increase or decrease the size of the virtual machine's memory.
- Increase or decrease the number of CPU cores per virtual socket.
- Add or remove CPUs.
- Assign a VDC compute policy.

You can also modify any of these configuration settings after the vApp is deployed. See Chapter 6 Reconfiguring vApps and Virtual Machines .

Before you create or update the virtual hardware configuration of a VM, retrieve the list of virtual hardware versions supported by the organization VDC where the VM will be deployed. See Retrieve a List of Supported Virtual Hardware Versions and Guest Operating Systems.

Prerequisites

- This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.
- Review the current configuration of the vApp and its virtual machines. Configuration parameters in the `VAppTemplate` such as its `NetworkConfigSection` and `LeaseSettingsSection` affect all virtual machines in the vApp. Configuration parameters for individual virtual machines are defined in `Vm` elements in the `VAppTemplate`. A virtual machine's network connections are defined in its `NetworkConnectionSection`, and its hardware configuration is defined in its `VirtualHardwareSection`. To view the configuration of a virtual machine in a vApp template, you can retrieve the template and examine the `VirtualHardwareSection` of that `Vm`, or you can download the OVF descriptor of the vApp template, as shown in Download an OVF Descriptor or OVA File .

Procedure

1. To change the `name`, `Description`, or `NeedsCustomization` properties of the virtual machine, add a `VmGeneralParams` element to the `SourcedItem`.
2. Examine the OVF descriptor of the template to determine the values that you can include in the `VirtualHardwareSection` of the `SourcedItem` element.
3. Include the `SourcedItem` element in an `InstantiateVAppTemplateParams` element.
4. Make a POST request to the `action/instantiateVAppTemplate` URL of the VDC.

   Supply the `InstantiateVAppTemplateParams` element as the request body.

Example: Modify Virtual Machine Hardware During vApp Template Instantiation

This `InstantiateVAppTemplateParams` request extends the request shown in Instantiate a vApp Template and Modify Virtual Machine Name, Description, and Storage Profile to include a `SourcedItem` element that makes several configuration changes in the virtual machine referenced at `https://vcloud.example.com/api/vAppTemplate/vm-4`.

- Sets the value of `NeedsCustomization` to true.
- Adds a virtual CPU and changes the value of `CoresPerSocket` to `2`. If you include a `CoresPerSocket` element, its value must be an integer multiple of the value of the existing `rasd:VirtualQuantity` of CPU items, or of the value you supply in `NumberOfCpus`. Modify the CPU Configuration of a Virtual Machine shows the original CPU configuration, and how to make this change by reconfiguring the virtual machine in a deployed vApp.
- Increases the capacity of the hard disk from 1GB to 10GB by including a `Disk` element that specifies a `Size` of `10240` for the disk that has a `rasd:InstanceID` value of `2000`. The value

you supply for Size is interpreted as megabytes. You can see the original disk configuration in Retrieve the Hard Disks and Controllers in a Virtual Machine . Modify the Hard Disk Configuration of a Virtual Machine shows how to make the same change by reconfiguring the virtual machine in a deployed vApp. If you include a Disk element, the value of its instanceId attribute must match the value in the rasd:InstanceID element of an existing Item that defines a virtual disk (RASD resource type 17). Disk capacity can be increased, but not decreased, for disks on SATA and SCSI controllers. The capacity of other disk types cannot be changed. Item elements that represent SATA disks have a vcloud:busType attribute with the value 20. Those that represent SCSI disks have a vcloud:busType attribute with the value 6.

- Assigns the VDC compute policy *policy_vmGroup3_on_cluster_2*.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/instantiateVAppTemplate
Content-Type: application/vnd.vmware.vcloud.instantiateVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Linux FTP server"
   deploy="true"
   powerOn="true"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Example FTP Server</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks</ovf:Info>
         <NetworkConfig networkName="vAppNetwork">
            <Configuration>
               <ParentNetwork href="https://vcloud.example.com/api/network/54"/>
               <FenceMode>bridged</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
      <LeaseSettingsSection
         type="application/vnd.vmware.vcloud.leaseSettingsSection+xml">
         <ovf:Info>Lease Settings</ovf:Info>
         <StorageLeaseInSeconds>172800</StorageLeaseInSeconds>
         <StorageLeaseExpiration>2014-04-25T08:08:16.438-07:00</StorageLeaseExpiration>
      </LeaseSettingsSection>
   </InstantiationParams>
   <Source href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111"/>
   <SourcedItem>
      <Source href="https://vcloud.example.com/api/vAppTemplate/vm-4"/>
      <VmGeneralParams>
         <Name>ftp1</Name>
         <Description>Primary FTP Server Instance</Description>
         <NeedsCustomization>true</NeedsCustomization>
      </VmGeneralParams>
      <InstantiationParams>
```

```
         <ovf:VirtualHardwareSection
            xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
            xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
            xmlns:vmw="http://www.vmware.com/schema/ovf"
            xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
            xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_VirtualSystemSettingData"
            ovf:transport=""
            vcloud:href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/"
            vcloud:type="application/vnd.vmware.vcloud.virtualHardwareSection+xml">
            <ovf:Info>Virtual hardware requirements</ovf:Info>
            <ovf:Item>
               <rasd:AddressOnParent>0</rasd:AddressOnParent>
               <rasd:Description>Hard disk</rasd:Description>
               <rasd:ElementName>Hard disk 1</rasd:ElementName>
               <rasd:HostResource
                  xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
                  vcloud:capacity="10240"
                  vcloud:busSubType="lsilogicsas"
                  vcloud:busType="6"></rasd:HostResource>
               <rasd:InstanceID>2000</rasd:InstanceID>
               <rasd:ResourceType>17</rasd:ResourceType>
            </ovf:Item>
            <ovf:Item>
               <rasd:AllocationUnits>hertz * 10^6</rasd:AllocationUnits>
               <rasd:Description>Number of Virtual CPUs</rasd:Description>
               <rasd:ElementName>1 virtual CPU(s)</rasd:ElementName>
               <rasd:InstanceID>41</rasd:InstanceID>
               <rasd:Reservation>0</rasd:Reservation>
               <rasd:ResourceType>3</rasd:ResourceType>
               <rasd:VirtualQuantity>2</rasd:VirtualQuantity>
               <rasd:Weight>0</rasd:Weight>
               <vmw:CoresPerSocket>2</vmw:CoresPerSocket>
            </ovf:Item>
         </ovf:VirtualHardwareSection>
      </InstantiationParams>
      <StorageProfile href="https://vcloud.example.com/api/vdcStorageProfile/33">
      <VdcComputePolicy
         href="https://vcloud.example.com/cloudapi/vdcComputePolicies/
urn:vcloud:vdcComputePolicy:74"
         id="urn:vcloud:vdcComputePolicy:74"
         name="policy_vmGroup3_on_cluster_2"
         type="application/vnd.vmware.vcloud.vdcComputePolicyType+xml"/>
      </StorageProfile>
   </SourcedItem>
   <AllEULAsAccepted>true</AllEULAsAccepted>
</InstantiateVAppTemplateParams>
```

The response is a sparsely populated `VApp` element, as shown in the response portion of Deploying a vApp.

Compose a vApp From Existing Virtual Machines

With the VMware Cloud Director API composeVApp operation, you can build a vApp from existing virtual machines to which you have access.

Every VDC includes a link to a composeVApp operation, which creates a new vApp in it. The ComposeVappParams request body is a superset of InstantiateVAppTemplateParams, and composeVApp can generally be used wherever you would use instantiateVAppTemplate. To compose a vApp, POST a composeVApp request to this link. The request body is a ComposeVAppParams element, which includes the following information:

- An InstantiationParams element that can include any of the section types listed under Instantiation Parameters for vApps. This is where you define the vApp network to which all the virtual machines in the composed vApp connect, and custom vApp lease settings and startup parameters for the virtual machines.
- An optional Description of the composed vApp.
- Zero or more SourcedItem elements, each of which must contain a Source element that specifies the href of a Vm, VApp, or VAppTemplate to include in the composition. If the Source element references a virtual machine, the SourcedItem can include any of the following elements:
  - An InstantiationParams element specific to that virtual machine. This element can include any of the section types listed under Instantiation Parameters for Virtual Machines. Changes to most Item elements in a VirtualHardwareSection are ignored by the composeVApp operation.
  - A NetworkAssignment element that specifies how the network connections in the virtual machine are mapped to vApp networks defined in the InstantiationParams element that applies to the composed vApp.
  - A VAppScopedLocalId element that provides a unique identifier for the virtual machine in the scope of the composed vApp.

  If the Source element references a vApp or vApp template, all Vm elements from each composition source become peers in the Children collection of the composed vApp.
- If any of the composition items is subject to a EULA, the ComposeVAppParams element must include an AllEULAsAccepted element that has a value of true, indicating that you accept the EULA. Otherwise, composition fails.

The composed vApp must be deployed and powered on before you can use it.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Find the `composeVApp` link in the target VDC.

The XML representation of the VDC contains a `composeVapp` link, which has the following form:

```
<Link
   rel="add"
   type="application/vnd.vmware.vcloud.composeVAppParams+xml"
   href="https://vcloud.example.com/api/vdc/5/action/composeVApp" />
```

2 Create a `ComposeVappParams` element that specifies the details of the composition. See Compose a vApp

3 POST the `ComposeVappParams` element to the `composeVapp` link of the target VDC.

See the Request portion of Compose a vApp.

Example: Compose a vApp

A `ComposeVAppParams` request body includes a root-level `InstantiationParams` element that provides instantiation parameters for the composed vApp. The request body can include an arbitrary number of `SourcedItem` elements, each of which can specify a vApp template or a virtual machine. `SourcedItem` elements where the `Source` is a vApp template cannot contain `InstantiationParams`. If you want to modify any of the virtual machines during composition or recomposition, specify `InstantiationParams` for them in the containing `Source` element.

```
<!- ComposeVAppParams/RecomposeVAppParams request body prototype -->
<ComposeVAppParams>
   <InstantiationParams>
      <!-- vApp-level params -->
   </InstantiationParams>
   <SourcedItem>
      <Source href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-111">
         <!-- InstantiationParams not allowed if Source references a vApp template -->
         <!-- vApp-level InstantiationParams apply to the composed vApp -->
   </SourcedItem>
   <SourcedItem>
      <Source href="any Vm">
         <InstantiationParams>
            <!-- Vm-level params -->
         </InstantiationParams>
      </Source>
   </SourcedItem>
</ComposeVAppParams>
```

This request composes a vApp from two virtual machines. The two `SourcedItem` elements each define a virtual machine (in their `Source` element) and supply `InstantiationParams` that modify its `NetworkConnectionSection` to connect to the vApp network created for this vApp in the root level `InstantiationParams` element.

Note If a virtual machine referenced in a `Source` element is powered on or has an independent disk attached, this operation will fail. You can use a query like this one to return a list of references to powered-off virtual machines that you have access to.

```
https://vcloud.example.com/api/query?type=adminVM&format=references&filter=status==POWERED_OFF
```

See Chapter 10 Using the Query Service.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/composeVApp
Content-Type: application/vnd.vmware.vcloud.composeVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ComposeVAppParams
   name="Example Corp’s CRM Appliance"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Composed CRM Appliance</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks</ovf:Info>
         <NetworkConfig networkName="CRMApplianceNetwork">
            <Configuration>
               <ParentNetwork href="https://vcloud.example.com/api/network/54"/>
               <FenceMode>natRouted</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
   </InstantiationParams>
   <SourcedItem>
      <Source href="https://vcloud.example.com/api/vApp/vm-4"/>
      <InstantiationParams>
         <NetworkConnectionSection
            xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
            type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
            href="https://vcloud.example.com/api/vApp/vm-4/networkConnectionSection/"
            ovf:required="false">
            <ovf:Info/>
            <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
            <NetworkConnection network="CRMApplianceNetwork">
               <NetworkConnectionIndex>0</NetworkConnectionIndex>
               <IsConnected>true</IsConnected>
               <IpAddressAllocationMode>DHCP</IpAddressAllocationMode>
            </NetworkConnection>
         </NetworkConnectionSection>
      </InstantiationParams>
   </SourcedItem>
   <SourcedItem>
      <Source href="https://vcloud.example.com/api/vApp/vm-45"/>
      <InstantiationParams>
         <NetworkConnectionSection
            xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
            type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
            href="https://vcloud.example.com/api/vApp/vm-4/networkConnectionSection/"
            ovf:required="false">
            <ovf:Info/>
            <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
            <NetworkConnection network="CRMApplianceNetwork">
               <NetworkConnectionIndex>0</NetworkConnectionIndex>
               <IsConnected>true</IsConnected>
               <IpAddressAllocationMode>DHCP</IpAddressAllocationMode>
            </NetworkConnection>
         </NetworkConnectionSection>
      </InstantiationParams>
   </SourcedItem>
   <AllEULAsAccepted>true</AllEULAsAccepted>
</ComposeVAppParams>
```

The response is a sparsely populated `VApp` element in the target VDC. When the `Task` embedded in the response is complete, the vApp has been composed.

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
   name="Example Corp's CRM Appliance"
   type="application/vnd.vmware.vcloud.vApp+xml"
   status="8"
   href="https://vcloud.example.com/api/vApp/vapp-33" ...>
   <Link
      rel="up" type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"/>
   ...
   <Description>Composed CRM Appliance</Description>
   ...
   <Tasks>
      <Task operation="Composing Virtual Application Example Corp's CRM Appliance (33)" ...>
         ...
      </Task>
   </Tasks>
</VApp>
```

Recompose a vApp to Add, Remove, or Reconfigure Virtual Machines

The VMware Cloud Director API supports recomposition of a vApp to add, remove, and reconfigure its virtual machines. To recompose a vApp, make a `recomposeVApp` request, supplying a `RecomposeVAppParams` element as the request body.

The `RecomposeVAppParams` element allows an arbitrary number of `DeleteItem` and `ReconfigureItem` elements, but is otherwise identical to `ComposeVAppParams`.

Unlike a `composeVapp` request, which operates on a VDC and creates a new vApp, a `recomposeVapp` request operates on (and modifies) an existing vApp. The XML representation of a vApp contains a `recomposeVApp` link, which has the following form:

```
<Link
   rel="recompose"
   type="application/vnd.vmware.vcloud.recomposeVAppParams+xml"
   href="https://vcloud.example.com/api/vApp/vapp-33/action/recomposeVApp" />
```

To recompose a vApp, POST a `recomposeVApp` request to this link. The request body is a `RecomposeVAppParams` element, which can include all the information allowed in a `ComposeVAppParams`, as well as the following additional elements:

- `ReconfigureItem` elements that reconfigure virtual machines in the vApp. By including `ReconfigureItem` elements, you can enable a `recomposeVApp` request to make the same kinds of virtual machine configuration changes you can make with a `reconfigureVm` request. See Update Multiple Sections of a Virtual Machine
- `DeleteItem` elements remove virtual machines from the vApp

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Find the `recomposeVApp` link in the target vApp.

2 Create a `RecomposeVappParams` element that specifies the details of the recomposition.

See Recompose a vApp.

3 POST the `RecomposeVappParams` element to the `recomposeVapp` link of the target vApp.

Example: Recompose a vApp

A RecomposeVAppParams request body prototype is similar to the ComposeVAppParams prototype shown in Compose a vApp. Instantiation parameters for the recomposed vApp appear as a root level InstantiationParams element. Instantiation parameters for added virtual machines appear as InstantiationParams in SourcedItem elements whose Source element references a virtual machine.

This example uses the recomposeVApp operation to modify this vApp, which contains three virtual machines. Only a few of the elements in the vApp appear here.

```
<VApp
   name="Example Corp’s CRM Appliance"
   type="application/vnd.vmware.vcloud.vApp+xml"
   status="8"
   href="https://vcloud.example.com/api/vApp/vapp-33" ...>
   ...
   <Link
      rel="recompose"
      type="application/vnd.vmware.vcloud.recomposeVAppParams+xml"
      href="https://vcloud.example.com/api/vApp/vapp-33/action/recomposeVApp" />
   ...
   <Children>
      <Vm
         status="8"
         name="CRM-DB-ORCL"
         href="https://vcloud.example.com/api/vApp/vm-7" ...>
         ...
      </Vm>
      <Vm
         status="8"
         name="CRM-CRM"
         href="https://vcloud.example.com/api/vApp/vm-44" ...>
         ...
      </Vm>
      <Vm
         status="8"
         name="CRM-HTTP"
         href="https://vcloud.example.com/api/vApp/vm-45" ...>
         ...
      </Vm>
   </Children>
   ...
</VApp>
```

Imagine that you want to replace the database server for this vApp with a new one hosted by a virtual machine that has the following properties:

```
<Vm
   status="8"
   name="CRM-DB-POSTGRES"
   href="https://vcloud.example.com/api/vApp/vm-90" ...>
   ...
</Vm>
```

To do that, create a request like this one, which removes the old database server (`DeleteItem`), adds the new one (`SourcedItem`, including InstantiationParams that connect it to the vApp network), and modifies the `StartupSection` to specify an appropriate start order for the recomposed vApp.

Request:

```
POST https://vcloud.example.com/api/vApp/vapp-33/action/recomposeVApp
Content-Type: application/vnd.vmware.vcloud.recomposeVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<RecomposeVAppParams
   name="Example Corp’s CRM Appliance"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Composed CRM Appliance</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks</ovf:Info>
         <NetworkConfig networkName="CRMApplianceNetwork">
            <Configuration>
               <ParentNetwork href="https://vcloud.example.com/api/network/54"/>
               <FenceMode>natRouted</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
      <ovf:StartupSection
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:type="application/vnd.vmware.vcloud.startupSection+xml">
         <ovf:Info>VApp startup section</ovf:Info>
         <ovf:Item
            ovf:order="0"
            ovf:id="CRM-DB-POSTGRES"/>
         <ovf:Item
            ovf:order="1"
            ovf:id="CRM-CRM"/>
      </ovf:StartupSection>
   </InstantiationParams>
   <SourcedItem>
      <Source href="https://vcloud.example.com/api/vApp/vm-90"/>
      <InstantiationParams>
         <NetworkConnectionSection
            xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
            type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
            href="https://vcloud.example.com/api/vApp/vm-4/networkConnectionSection/"
            ovf:required="false">
            <ovf:Info/>
            <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
            <NetworkConnection network="CRMApplianceNetwork">
               <NetworkConnectionIndex>0</NetworkConnectionIndex>
               <IsConnected>true</IsConnected>
               <IpAddressAllocationMode>DHCP</IpAddressAllocationMode>
            </NetworkConnection>
         </NetworkConnectionSection>
      </InstantiationParams>
   </SourcedItem>
   <AllEULAsAccepted>true</AllEULAsAccepted>
   <DeleteItem href="https://vcloud.example.com/api/vApp/vm-8"/>
</RecomposeVAppParams>
```

If the virtual machine referenced in `SourceItem` element in this example was already included in the vApp but needed to be reconfigured to adapt to the new vApp network configuration, you could use a `ReconfigureItem` element in place of the `SourcedItem` element.

```
<?xml version="1.0" encoding="UTF-8"?>
<RecomposeVAppParams
   name="Example Corp’s CRM Appliance"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>Composed CRM Appliance</Description>
   <InstantiationParams>
      ...
   </InstantiationParams>
   <AllEULAsAccepted>true</AllEULAsAccepted>
   <DeleteItem href="https://vcloud.example.com/api/vApp/vm-8"/>
   <ReconfigureItem
      name="vm-90"
      href="https://vcloud.example.com/api/vApp/vm-90">
      <ovf:Section
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:type="NetworkConnectionSectionType">
         <ovf:Info>My Network Connection Section Info</ovf:Info>
         <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
         <NetworkConnection network="CRMApplianceNetwork">
            <NetworkConnectionIndex>0</NetworkConnectionIndex>
            <IsConnected>true</IsConnected>
            <IpAddressAllocationMode>DHCP</IpAddressAllocationMode>
         </NetworkConnection>
      </ovf:Section>
   </ReconfigureItem>
</RecomposeVAppParams>
```

Note If a virtual machine referenced in a `Source` element is powered on or has an independent disk attached, this operation will fail. You can use a query like this one to return a list of references to powered-off virtual machines that you have access to.

```
https://vcloud.example.com/api/query?type=adminVM&format=references&filter=status==POWERED_OFF
```

See Chapter 10 Using the Query Service.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ...
   operation="Updating Virtual Application Example Corp’s CRM Appliance (33)" ...>
...
</Task>
```

Clone a vApp

You can make a copy of a vApp by cloning it. If the vApp is deployed when you clone it, the clone procedure also clones the memory state of the virtual machines in the vApp.

The `cloneVApp` request makes a copy of the vApp referenced in the `Source` element of the `CloneVappParams` request body. The request specifies a new name and, optionally, a new description for the copy.

The request can optionally include an `IsSourceDelete` element whose value specifies whether to delete the source vApp after the copy is complete. If `IsSourceDelete` is missing from the request body, or present with a value of `false`, the source object remains in place after the copy is complete. Setting `IsSourceDelete` to true effectively moves or renames the vApp.

The request can also include a `SourcedItem` element for each VM that you want to modify during the cloning process. In each `SourcedItem` element, you must specify the target VM in a `Source` element and the new values of the properties that you want to modify. For example, if you want to configure a new storage profile for a cloned VM, use the `StorageProfile` element in the `SourcedItem` element. If you want to configure a VDC compute policy for a cloned VM, use the `VdcComputePolicy` element in the `SourcedItem` element. For information about VDC compute policies, see Managing VM-Host Affinity Rules.

If the vApp is deployed when you clone it and the target VDC is backed by the same provider VDC as the source VDC, the clone is created with the following properties:

- Memory state of all virtual machines in the source vApp is preserved in the clone.

- The clone is suspended and connected to an isolated network.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the VDC in which you want to deploy the cloned vApp.

   Use a request like this one, where *id* is the identifier of the VDC:

   ```
   GET https://vcloud.example.com/vdc/id
   ```

   You can create the clone in the same VDC that holds the source vApp, or in a different VDC.

2 Examine the response to locate the `Link` element that contains the URL for cloning a vApp.

   This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.vcloud.cloneVAppParams+xml`, as shown here:

   ```
   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.cloneVAppParams+xml"
      href="https://vcloud.example.com/api/vdc/12/action/cloneVApp"/>
   ```

3 Create a `CloneVappParams` element that references the vApp to clone and specifies details of the clone operation.

   See Clone a vApp.

4 POST the `CloneVappParams` element to the `action/cloneVApp` link of the VDC in which you want to create the clone.

Example: Clone a vApp

This request creates a copy of the vApp created in Instantiate a vApp Template and Modify Virtual Machine Name, Description, and Storage Profile in another VDC. Because the `ParentNetwork` in the `Source` vApp is not available in the VDC specified in the `action/cloneVApp` request, the `CloneVAppParams` request body must include `InstantiationParams` that specify a new `ParentNetwork` for the vApp network, one that is available in the target VDC. The request also includes a `SourcedItem` specifying a new storage profile and a VDC compute policy for the virtual machine in the vApp.

Note If the vApp is deployed when you clone it, any network configuration you specify in the `CloneVAppParams` is ignored and the clone is created with a connection to an isolated network.

Request:

```
POST https://vcloud.example.com/api/vdc/12/action/cloneVApp
Content-Type: application/vnd.vmware.vcloud.cloneVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CloneVAppParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   name="cloned"
   deploy="false"
   powerOn="false">
   <Description>Cloned vApp Example</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks</ovf:Info>
         <NetworkConfig networkName="vAppNetwork">
            <Configuration>
               <ParentNetwork href="https://vcloud.example.com/api/network/101"/>
               <FenceMode>bridged</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
   </InstantiationParams>
   <Source href="https://vcloud.example.com/api/vApp/vapp-7"/>
   <IsSourceDelete>false</IsSourceDelete>
   <SourcedItem>
      <Source href="https://vcloud.example.com/api/vAppTemplate/vm-4"/>
      <StorageProfile href="https://vcloud.example.com/api/vdcStorageProfile/45">
      </StorageProfile>
      <VdcComputePolicy href="https://vcloud.example.com/cloudapi/vdcComputePolicies/74"
name="policy_vmGroup2_cluster1">
      </VdcComputePolicy>
   </SourcedItem>
</CloneVAppParams>
```

The response is a sparsely populated `VApp` element in the target VDC. When the `Task` embedded in the response is complete, the vApp has been cloned.

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   deployed="false"
   status="0"
   name="cloned"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-103">
   ...
   <Link
```

```
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/12"/>
   ...
   <Description>Cloned vApp Example</Description>
   ...
   <Tasks>
      <Task
         status="running"
         operation="Copying Virtual Application cloned (103)"
         ... >
      </Task>
   </Tasks>
   ...
</VApp>
```

Create a vApp From an OVF Package

An `instantiateOvf` request uploads an OVF package and then creates a vApp from it. By default, this operation also deploys the vApp and powers it on.

If you want to deploy an OVF package as a vApp without creating a vApp template and corresponding catalog item, make an `instantiateOvf` request. This request initiates a workflow similar to the one shown in Upload an OVF Package to Create a vApp Template , but with a few important differences.

- The target of the upload is a VDC, not a catalog.
- The request body is an `InstantiateOvfParams` element, not an `UploadVAppTemplateParams` element.
- The response is a `VApp` element that includes an upload URL for the OVF descriptor.

After you retrieve the `VApp` element created by the `instantiateOvf` request, you can upload the descriptor and the files it references.

Prerequisites

Verify that the following are true:

- You have an OVF package to upload.
- You are logged in as a user who has permission to upload OVF packages and create vApps.
- You know the URL of the target VDC that will receive the upload. Retrieve the XML representation of your organization to see a list of the VDCs that it contains.

Review the contents of the Envelope element in the descriptor file of your OVF package. Several properties in this file have implications for the `InstantiateOvfParams` request body you must construct to initiate the upload.

Procedure

1 Retrieve the XML representation of the target VDC.

2 Examine the response to locate the `Link` element that contains the URL for creating a vApp from an OVF package.

   This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.vcloud.instantiateOvfParams+xml`, as shown here:

```
<Link
   rel="add"
   type="application/vnd.vmware.vcloud.instantiateOvfParams+xml"
   href="https://vcloud.example.com/api/vdc/44/action/instantiateOvf" />
```

3 Create an `InstantiateOvfParams` request body.

   See the request portion of Create a vApp From an OVF Package .

4 POST the `InstantiateOvfParams` to the `instantiateOvf` URL you retrieved in Step 2

   See the request portion of Create a vApp From an OVF Package .

5 Examine the response.

   The response, a `VApp` element, contains a `File` element that specifies an upload URL for the OVF descriptor. See the response portion of Create a vApp From an OVF Package

6 Upload the OVF descriptor.

   Make a PUT request to the upload URL in the `VApp`. The upload URL for the OVF descriptor is in a `Link` element with the following form:

```
<Link
   rel="upload:default"
   href="https://vcloud.example.com/transfer/.../descriptor.ovf" />
```

   Supply the OVF descriptor as the request body. The OVF descriptor contains a single `Envelope` element.

7 Retrieve the remaining upload URLs

   a Retrieve the `VApp`.

   b Verify that the value of the `ovfDescriptorUploaded` attribute is `true`.

   c Examine the `VApp` to find the upload URLs for the files referenced in the OVF descriptor.

      These URLs are contained in `Link` elements where `rel="upload:default"`.

8 Upload the referenced files.

   You can follow the progress of the upload by retrieving the vApp and noting the progress of the embedded `Task`.

Results

After all of the referenced files are uploaded, the VApp element no longer includes an embedded Task. The vApp is placed in the power and deployment state that the values of the powerOn and deploy attributes specify in the request body.

Example: Create a vApp From an OVF Package

This request includes a NetworkMapping element that maps a network name found in the uploaded OVF descriptor to the name of a network available in the target VDC.

Request:

```
POST https://vcloud.example.com/api/vdc/5/action/instantiateOvf
Content-Type: application/vnd.vmware.vcloud.instantiateOvfParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateOvfParams
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="W2K8">
   <Description>Example vApp</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks
         </ovf:Info>
         <NetworkConfig
            networkName="vAppNetwork">
            <Configuration>
               <ParentNetwork
                  href="https://vcloud.example.com/api/network/54" />
               <FenceMode>bridged</FenceMode>
            </Configuration>
         </NetworkConfig>
      </NetworkConfigSection>
   </InstantiationParams>
   <AllEULAsAccepted>true</AllEULAsAccepted>
   <NetworkMapping>
      <Source>Network 1</Source>
      <Target>vAppNetwork</Target>
   </NetworkMapping>
   <InstantiateVmParams
      id="VM-1">
      <Name>VM-1</Name>
      <NetworkConnectionSection>
         <ovf:Info />
         <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
         <NetworkConnection
            network="Network 1">
            <NetworkConnectionIndex>0</NetworkConnectionIndex>
            <IsConnected>true</IsConnected>
            <IpAddressAllocationMode>POOL</IpAddressAllocationMode>
         </NetworkConnection>
      </NetworkConnectionSection>
      <ComputerName>W2K8</ComputerName>
   </InstantiateVmParams>
</InstantiateOvfParams>
```

The response is a sparsely populated `VApp` element that includes an upload URL for the OVF descriptor. See Uploading Referenced Files for file upload procedures.

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   deployed="false"
   status="0"
   name="W2K8"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-23">
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/5"/>
   <Description>Example vApp</Description>
      <Link
      rel="remove"
      href="https://vcloud.example.com/api/vApp/vapp-23" />
   <Description>Example vApp</Description>
   <Tasks>
      <Task
         status="running"
         operation="Creating Virtual Application W2K8(23)"
         ... >
         ...
      </Task>
   </Tasks>
   <Files>
      <File
         name="descriptor.ovf"
         bytesTransferred="0">
         <Link
            rel="upload:default"
            href="https://vcloud.example.com/transfer/.../descriptor.ovf" />
      </File>
   </Files>
   <Owner>
      ...
   </Owner>
   <InMaintenanceMode>false</InMaintenanceMode>
</VApp>
```

Capture a vApp as a Template

You can capture a vApp to create a vApp template in a catalog. Instantiating the resulting template recreates the vApp from which it was captured.

The `captureVApp` request creates a template based on the vApp referenced in the `Source` element of the `CaptureVAppParams` request body. The request specifies a new name and, optionally, a new description for the template.

The request can optionally specify a VDC compute policy for particular VMs in the vApp template. You can include a `CaptureVmParams` element for each VM for which you want to configure a VDC compute policy. In the `CaptureVmParams` element, you must specify the target VM in a `Source` element and the target VDC compute policy in a `VdcComputePolicy` element. Instantiating the template assigns the configured VDC compute policy to the newly created VMs in the instantiated vApp. By configuring VDC compute policies, you can restrict the VDCs in which users can instantiate the vApp template. For example, if you create *oracle_vdc_compute_policy* and publish it only to *vdc1*, when you configure this VDC compute policy in a vApp template and disable customization on instantiation, users cannot instantiate this template in VDCs other than *vdc1*. For information about VDC compute policies, see Managing VM-Host Affinity Rules.

If you want the new template to overwrite an existing template in the catalog, you can specify a `TargetCatalogItem` element in the request. Otherwise, the new template is stored in a new catalog item.

If the vApp is deployed when you capture it, the template is created with the following properties.

- Memory state of all virtual machines in the source vApp is preserved in the template.
- Instantiating the template always creates a suspended vApp connected to an isolated network.

Note If the template is instantiated in a VDC that is not backed by the provider VDC that backed the VDC in which the vApp was captured, memory state in the template is discarded on instantiation, and the vApp is created with the network connections defined in the template or instantiation parameters.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that the target catalog does not have an external subscription.

Procedure

1 Retrieve the XML representation of the catalog to which to add the vApp template that the capture creates.

Use a request like this one, where *id* is the identifier of the catalog:

```
GET https://vcloud.example.com/api/catalog/id
```

2 Examine the response to locate the `Link` element that contains the URL for capturing a vApp.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.vcloud.captureVAppParams+xml`, as shown here:

```
<Link
   rel="add"
   type="application/vnd.vmware.vcloud.captureVAppParams+xml"
   href="https://vcloud.example.com/api/catalog/5/action/captureVApp"/>
```

3 Create a `CaptureVAppParams` element that references the vApp to capture.

See Capture a vApp.

4 POST the `CaptureVAppParams` element to the `action/captureVApp` link shown in Step 2.

Example: Capture a vApp

This request captures the vApp created in Compose a vApp and configures the VCD compute policy `oracle_policy` for two of the containing VMs. Because the request does not specify a `TargetCatalogItem`, a new catalog item is created for the new template.

Request:

```
POST https://vcloud.example.com/api/catalog/5/action/captureVApp
Content-Type: application/vnd.vmware.vcloud.captureVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CaptureVAppParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Example Corp’s CRM Appliance">
   <Description>Captured CRM Appliance</Description>
   <Source
      href="https://vcloud.example.com/api/vApp/vapp-33" />
   <CaptureVmParams>
         <Source
      href="https://vcloud.example.com/api/vApp/vm-4" />
         <VdcComputePolicy
         href="https://vcloud.example.com/cloudapi/vdcComputePolicies/74"
         name="oracle_policy" />
   </CaptureVmParams>
   <CaptureVmParams>
      <Source
         href="https://vcloud.example.com/api/vApp/vm-5" />
      <VdcComputePolicy
```

```
         href="https://vcloud.example.com/cloudapi/vdcComputePolicies/74"
         name="oracle_policy" />
   </CaptureVmParams>
</CaptureVAppParams>
```

The response is a sparsely populated `VApp` element in the target VDC. It contains a `Link` to the catalog specified in the request. When the `Task` embedded in the response is complete, the vApp has been captured and a vApp template created in the target catalog.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-232">
   ...
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/5" />
      ...
   <Description>Captured CRM Appliance</Description>
   ...
   <Tasks>
      <Task
         ...
         operation="Capturing Virtual Application Template Example Corp’s CRM Appliance (232)"
         ...
        </Task>
   </Tasks>
   ...
</VAppTemplate>
```

Update vApp Access Controls

An administrator or the vApp owner can use the `controlAccess` links in a `VApp` element to grant or restrict access to the vApp.

A vApp initially grants full access to its owner and no access to other users. The VMware Cloud Director API access control mechanism enables an administrator to retrieve or update vApp access controls to add or remove rights for all users, or for individual users. For a general discussion of access controls in VMware Cloud Director, see Controlling Access to vApps and Catalogs.

Prerequisites

Verify that you are logged in to the VMware Cloud Director API as an administrator or the object owner.

Procedure

1 Retrieve the XML representation of the vApp.

  Use a request like this one:

  ```
  GET https://vcloud.example.com/api/vApp/vapp-7
  ```

2 Examine the `VApp` element to find the `controlAccess` links that it contains.

3 Create a `ControlAccessParams` element request body that specifies the details of the update.

4 POST the `ControlAccessParams` element to the `action/controlAccess` link that you retrieved in Step 1.

Example: Update vApp Access Controls

This request updates the access controls of a vApp to grant full control to one user and read-only access to another user. The request body, a `ControlAccessParams` element, specifies a value of `false` for the `IsSharedToEveryone` element, and contains an `AccessSetting` element for each user whose access rights are being modified. Each user is identified by a reference to a `User` object. See User and Group Administration. The response, a subset of which appears in this example, echoes the request.

Request:

```
POST https://vcloud.example.com/api/vApp/vapp-7/action/controlAccess
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/40"/>
         <AccessLevel>FullControl</AccessLevel>
      </AccessSetting>
      <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/45"/>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
   </AccessSettings>
</ControlAccessParams>
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      ...
   </AccessSettings>
</ControlAccessParams>
```

Create a VM-VM Affinity Rule

A VM-VM affinity rule applies to two or more virtual machines and specifies whether they should be deployed on the same host or on separate hosts. VM-VM affinity rules are properties of an organization VDC, and apply to virtual machines deployed on the hosts that back that VDC.

VM-VM affinity rules are user-specified constraints that the system considers during the deployment process. There are two types, or polarities, of VM-VM affinity rules:

- A rule with a `Polarity` value of `Affinity` specifies two or more virtual machines that the system should deploy on the same host. This kind of rule is commonly applied in cases where virtual machines can realize performance benefits from being placed on a single host and are running workloads where failure of that host can be tolerated.
- A rule with a `Polarity` value of `Anti-Affinity` specifies two or more virtual machines that the system should deploy on separate hosts. This kind of rule is commonly applied in cases where virtual machines are intended to support redundancy, availability, and similar uses that must not be affected by the failure of one host.

You can make an affinity rule mandatory to provide a hint to the system that the deployment should fail unless the specified placement objective can be achieved. When a virtual machine is the subject of a mandatory affinity rule, placement requirements dictated by the rule override any placement changes that might be initiated by vSphere services such as High Availability and Storage DRS.

When you create a VM-VM affinity rule in an organization VDC, VMware Cloud Director creates a corresponding rule in the vCenter server that provides resources to that organization VDC. This rule has several differences compared with a VM-VM affinity rule created directly in vCenter:

- A virtual machine can be referenced in no more than one VMware Cloud Director affinity rule.
- A virtual machine can be referenced in no more than one VMware Cloud Director anti-affinity rule.

When you create a VMware Cloud Director affinity rule, the system validates that it does not reference a virtual machine that is already referenced by an existing rule of the same polarity. Requests that attempt to create a rule that would violate this constraint are rejected.

Prerequisites

- This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.
- To create an affinity rule, you must be the owner of all virtual machines specified in the rule.

Procedure

1. Retrieve the XML representation of the organization VDC to which you want to add the affinity rule.

   This request retrieves the user view of an organization VDC.

   ```
   GET https://vcloud.example.com/api/vdc/44
   ```

2. Examine the response to locate the `Link` element that contains the URL for adding affinity rules to the VDC.

   This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.vcloud.vmaffinityrule+xml`, as shown here:

   ```
   <Link
         rel="add"
         href="https://vcloud.example.com/api/vdc/44/vmAffinityRules/"
         type="application/vnd.vmware.vcloud.vmaffinityrule+xml" />
   ```

3. Retrieve references to the virtual machines that will be subject to the rule.

   The references must be to virtual machines that have been instantiated in the same VDC. You can use a query like this one to list references to all the virtual machines instantiated in a VDC whose URL is *vdc-reference*.

   ```
   GET https://vcloud.example.com/api/query?\
   type=vm&format=references&\
   filter=isVAppTemplate==false&vdc==vdc-reference
   ```

   For example:

   ```
   GET https://vcloud.example.com/api/query?\
   type=vm&format=references&\
   filter=isVAppTemplate==false&vdc==https://vcloud.example.com/api/vdc/44
   ```

4. Create a `VmAffinityRule` request body.

   The `name` element cannot contain more that 40 characters. See Create a VM-VM Affinity Rule.

5. POST the `VmAffinityRule` element you created in Step 4 to the URL for adding affinity rules the organization VDC.

   See the request portion of Create a VM-VM Affinity Rule.

Example: Create a VM-VM Affinity Rule

This request creates an affinity rule for two virtual machines.

Request:

```
POST https://vcloud.example.com/api/vdc/44/vmAffinityRules
Content-Type: application/vnd.vmware.vcloud.vmaffinityrule+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<VmAffinityRule xmlns="http://www.vmware.com/vcloud/v1.5">
   <Name>example-affinity-rule</Name>
   <IsEnabled>true</IsEnabled>
   <IsMandatory>false</IsMandatory>
   <Polarity>Affinity</Polarity>
   <Scope>Host</Scope>
   <VmReferences>
      <VmReference href="https://vcloud.example.com/api/vApp/vm-44" />
      <VmReference href="https://vcloud.example.com/api/vApp/vm-200" />
   </VmReferences>
</VmAffinityRule>
```

If either of the referenced virtual machines is already the subject of an affinity rule, the request fails with an error message indicating the reason for the failure.

The response is a `Task`. Note the `href` of the `Owner` element in the returned `Task`. This is the ID of the rule. You must use it when you delete or update the rule.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<Task
   ...
   operationName="affinityRuleCreate"
   ... >
   <Owner
      href="https://vcloud.example.com/api/vmAffinityRule/edf7c456-
a69c-48b1-89a8-4e9674448e01"
      name="example_name"
      type="application/vnd.vmware.vcloud.vmaffinityrule+xml" />
   ...
</Task>
```

Note When you create or update an anti-affinity rule, the system might need to create a place-holder virtual machine to satisfy the vCenter requirement that such rules include at least two virtual machines. This place-holder is visible to vCenter administrators, but is not visible in VMware Cloud Director. It is never powered on.

Update or Delete a VM-VM Affinity Rule

To update or remove an affinity rule, use the `edit` and `remove` links in the rule.

Prerequisites

- This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.
- To update or delete an affinity rule, you must be the owner of all virtual machines specified in the rule.
- To update or remove the rule, you must know the `href` value of the rule. This value is returned when you create the rule (see Create a VM-VM Affinity Rule). You can retrieve a list of all your affinity rules from a VDC by making a GET request to this link in the VDC.

```
<Link rel="down" href="https://vcloud.example.com/api/vdc/44/vmAffinityRules/"
    type="application/vnd.vmware.vcloud.vmaffinityrules+xml"/>
```

Procedure

1 Retrieve the XML representation of the rule.

Use a request like this one.

```
GET https://vcloud.example.com/api/vmAffinityRule/edf7c...
```

2 Examine the response to locate the `Link` elements that contain the URLs for modifying or deleting the rule.

```
<VmAffinityRule ...>
   ...
   <Link
      rel="edit"
      href="https://vcloud.example.com/api/vmAffinityRule/edf7c..."
      type="application/vnd.vmware.vcloud.vmaffinityrule+xml" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/vmAffinityRule/edf7c..." />
   ...
</VmAffinityRule>
```

3 Use one of these links to update or delete the rule.

Option	Description
To update the rule:	Make a PUT request to the rule's `edit` URL and supply a `VmAffinityRule` as the body. See Update a VM-VM Affinity Rule.
To delete the rule:	Make a DELETE request to the rule's `remove` URL.

Example: Update a VM-VM Affinity Rule

This request updates the an affinity rule created in Create a VM-VM Affinity Rule to add a third virtual machine.

Request:

```
PUT https://vcloud.example.com/api/vmAffinityRule/edf7c...
Content-Type: application/vnd.vmware.vcloud.vmaffinityrule+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<VmAffinityRule xmlns="http://www.vmware.com/vcloud/v1.5">
   <Name>example-affinity-rule</Name>
   <IsEnabled>true</IsEnabled>
   <Polarity>Affinity</Polarity>
   <VmReferences>
      <VmReference href="https://vcloud.example.com/api/vApp/vm-44" />
      <VmReference href="https://vcloud.example.com/api/vApp/vm-200" />
 <VmReference href="https://vcloud.example.com/api/vApp/vm-133" />
   </VmReferences>
</VmAffinityRule>
```

The response is a `Task`.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<Task
   ...
   operationName="affinityRuleUPdate"
   ... >
   <Owner
      href="https://vcloud.example.com/api/vmAffinityRule/edf7c456-
a69c-48b1-89a8-4e9674448e01"
      name="example_name"
      type="application/vnd.vmware.vcloud.vmaffinityrule+xml" />
   ...
</Task>
```

Specifying Advanced Virtual Machine Settings with `ExtraConfig` Elements

OVF `ExtraConfig` elements provide a flexible way of including *key*=*value* pairs in the configuration of a virtual machine. The keys and values are interpreted by the system when the virtual machine is deployed, and can be used to specify a variety of virtual machine properties.

The VMware Cloud Director API supports several `ExtraConfig` *key*=*value* pairs that you can use to configure virtual machines for specific types of workloads. You can use OVFtool to create an OVF package that includes virtual machines with these `ExtraConfig` values.

Permission to upload or download a `Vm` that includes any of these keys requires one or more of the following rights.

Table 5-2. `ExtraConfig` Keys, Values, and Required Rights

Key	Values	Required Right	Description
any key	any value	`vApp: Preserve All ExtraConfig Elements During OVF Import and Export`	A user with this right can upload or download an OVF package that contains an unlimited set of `ExtraConfig` *key=value* pairs. The set includes, but is not limited to, the pairs listed in this table.
any key that is a value of the `vapp.allowed.extra.config` system configuration property	any value	`vApp: Allow Matching Extra Config`	A user with this right can upload or download an OVF package that contains any of the `ExtraConfig` *key=value* pairs in which the *key* is a value of the `vapp.allowed.extra.config` system configuration property. See VMware Knowledge Base article https://kb.vmware.com/kb/2148573.
`sched.cpu.latencySensitivity`	high, normal (default)	`vApp: Preserve Latency ExtraConfig Elements During OVF Import and Export`	Set to `high` for virtual machines running latency-sensitive workloads.
`ethernet`*n*`.coalescingScheme` (where *n* is the number of the virtual NIC in the range 0-9)	enabled, disabled (default)	`vApp: Preserve Ethernet-Coalescing ExtraConfig Elements During OVF Import and Export`	Set to `enabled` and specify the virtual NICs that must disable interrupt coalescing.

Table 5-2. `ExtraConfig` Keys, Values, and Required Rights (continued)

Key	Values	Required Right	Description
`numa.nodeAffinity`	*n*, ..., where *n* is a NUMA node number on the host.	`vApp: Preserve NUMA Node Affinity ExtraConfig Elements During OVF Import and Export`	Constrains the set of NUMA nodes on which a virtual machine's virtual CPU and memory can be scheduled.
`sched.mem.lpage.enable1GPage`	true, false (default)	`vApp: Allow Latency Extra Config`	Set the key to `true` for virtual machines with 1 GB page size. **Important** Virtual machines with 1 GB page size are supported only for virtual datacenters with 100% memory resources guaranteed. A user with `vApp: Allow Latency Extra Config` right can: ■ import an OVF with 1 GB page size into VMware Cloud Director as a vApp or vApp template. ■ download a vApp or vApp template with 1 GB page size as an OVF or OVA. ■ instantiate from a vApp template with 1 GB page size ■ import and auto import a virtual machine with 1 GB page size from vCenter into VMware Cloud Director.

For detailed information about the effects of these settings, see *Best Practices for Performance Tuning of Telco and NFV Workloads in vSphere* (http://www.vmware.com/files/pdf/techpaper/vmware-tuning-telco-nfv-workloads-vsphere.pdf). For information about using the VMware Cloud Director API to add rights to a role, see Create a Role in Your Organization.

Example: Applying `ExtraConfig` Values to a Virtual Machine

This example shows how to use OVFtool to create an OVF package that contains a `VirtualHardwareSection` with an `ExtraConfig` setting.

```
<?xml version="1.0" encoding="UTF-8"?>
<Vm ...>
   ...
   <ovf:VirtualHardwareSection
      xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
      ovf:transport="">
      <ovf:Info>Virtual hardware requirements</ovf:Info>
      <vmw:ExtraConfig
         ovf:required="false"
         vmw:key="sched.cpu.latencySensitivity"
         vmw:value="high" />
      ...
   </ovf:VirtualHardwareSection>
   ...
</Vm>
```

You can log in to VMware Cloud Director as a user whose role includes the right `vApp: Preserve Latency ExtraConfig Elements During OVF Import and Export` and upload the OVF package that you created.

Operate a vApp

`vApp` and `Vm` elements include a number of `action` links. You can use these links to operate a vApp or one of its virtual machines by making requests such as power on, suspend, power off, undeploy, and so on.

Only those action links that are valid for the vApp or virtual machine in its current state are returned. For example, if a vApp is instantiated but not deployed, only the links for `deploy` and `remove` are returned. For a vApp that is powered on, links for all actions except `powerOn` are returned. Some requests apply only to vApps, some apply only to virtual machines (`Vm` objects), and some apply to both.

- A request made to the `power` URL of a `VApp` invokes the requested operation on each of the virtual machines in the `Children` element of the `VApp`, in the order specified in its `ovf:StartupSection` element. This element, if present, specifies a start order and related properties for each member of a `VirtualSystemCollection` (each `Vm` contained by the `Children` element). If the element is not present, all members are started up at the same time. The same logic applies to shutdown, reboot, and similar operations.
- A request made to the `power` URL of a `Vm` affects only that virtual machine.

Prerequisites

This operation requires the rights included in the predefined **vApp User** role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the vApp to find the `action` links it contains.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/vApp/vapp-7
   ```

2 POST a request to the URL that implements the desired action.

   Many of these requests do not require a body.

Results

The server takes the requested action and returns a `Task` element that tracks the progress of the request.

vApp Power States

Operations that change the power state of a vApp or any of its VMs can have different results when initiated from the VMware Cloud Director API than they do when initiated from the VMware Cloud Director HTML5 UI. Understanding these state transitions and the operations that cause them can help you manage vApp and VM resource consumption and readiness.

As seen in the VMware Cloud Director HTML5 UI, a vApp can be Stopped, Running, or Partially Running. A VM can have any of the following states:

- Powered Off
- Powered On
- Suspended
- Partially Powered On (deployed
- Partially Suspended (deployed)

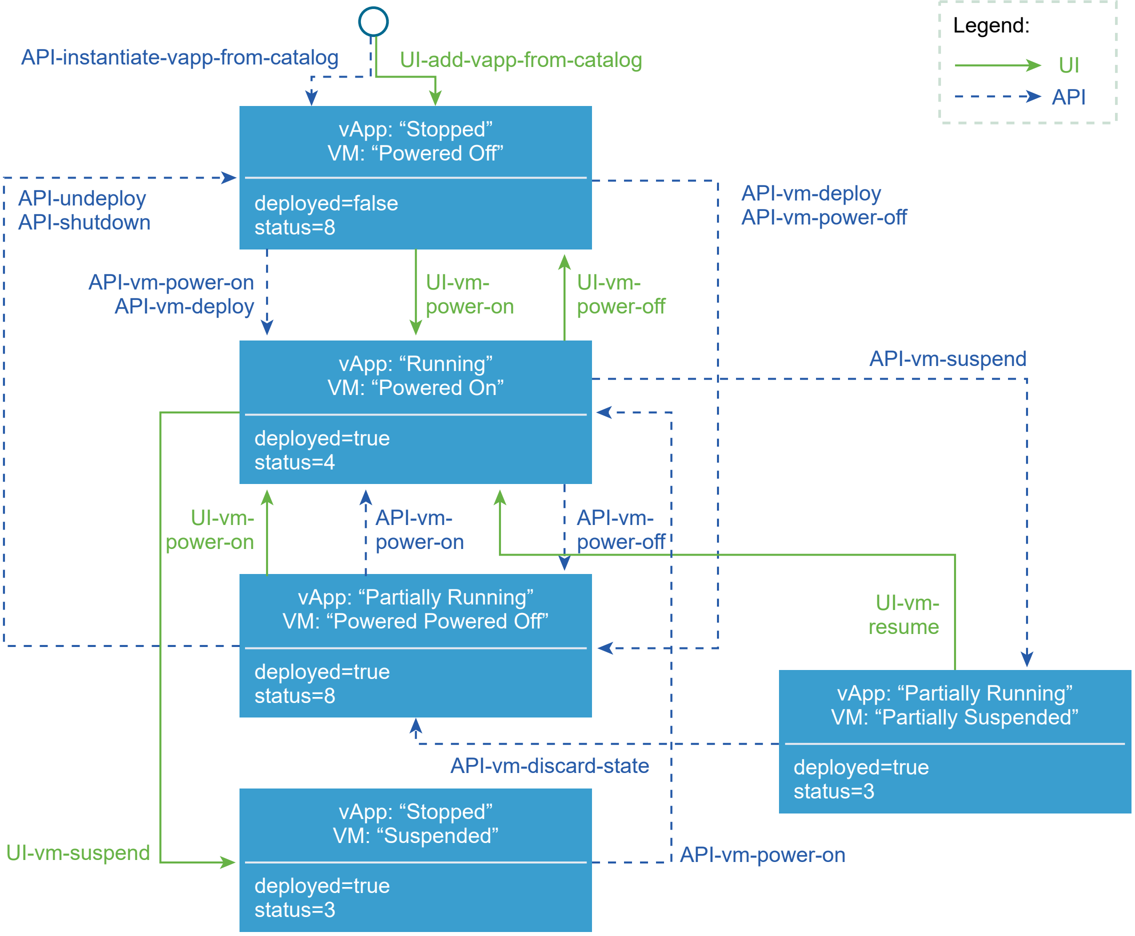
The VMware Cloud Director API returns information about VM power state in the `Vm` object’s status attribute:

Table 5-3. VM Power States and `status` Attribute Values

VM Power State	`Vm` Object `status` attribute value
Powered on	4
Powered off	8
Suspended	3

This simplified diagram covers most of the cases where a vApp contains a single VM. A vApp that contains multiple VMs can report its power state as **Partially Running** in the VMware Cloud Director HTML5 UI when some VMs are powered on and some are not.

Figure 5-2. vApp Power State Transitions



Provide User Input Requested by a Virtual Machine

A request for a virtual machine to change state (power on, suspend, reconfigure, and so on) might cause the virtual machine to ask for additional user input before it can complete.

A `vApp` that contains a `Vm` awaiting a user response has `status="5"`, and includes a link that you can GET to discover what input is needed.

Prerequisites

This operation requires the rights included in the predefined **vApp User** role or an equivalent set of rights.

Procedure

1 Find the `question` link in the target vApp.

This link has the following form:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.vmPendingQuestion+xml"
   href="http://vcloud.example.com/api/vApp/vm-5/question" />
```

2 Make a GET request to the URL in that link's href value.

The response is a `VmPendingQuestion` response that includes the question and the set of possible answers.

3 Create a `VmQuestionAnswer` element that supplies the answer.

See Provide User Input Requested by a Virtual Machine.

4 POST the element to the `question/action/answer` link included in the `VmPendingQuestion` response.

Example: Provide User Input Requested by a Virtual Machine

In this series of examples, a virtual machine that was recently reconfigured in vCenter to add a new parallel port device and then powered on is requesting user input about where to send output from the device. The `powerOn` request cannot complete until this input is supplied.

The first step is to use the `vmPendingQuestion` link, shown in Step 1, to get a `VmPendingQuestion` response that includes the question and the set of possible answers.

Request:

```
GET https://vcloud.example.com/api/vApp/vm-5/question
```

Response:

```
200 OK
Content-type: application/vnd.vmware.vcloud.vmPendingQuestion+xml
...
<VmPendingQuestion
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Link
      rel="answer"
      type="application/vnd.vmware.vcloud.vmPendingAnswer+xml"
      href="http://vcloud.example.com/api/vApp/vm-5/question/action/answer" />
   <Question>msg.parallel.file.open:Parallel port output file
      "/vmfs/volumes/d6162a46-58e50cab/linuxftp/vm-mgi.log" already
      exists. Do you want to replace it with any newly created content,
      or append new content to the end of the file?
</Question>
   <QuestionId>50</QuestionId>
   <Choices>
      <Id>0</Id>
      <Text>Append</Text>
   </Choices>
   <Choices>
      <Id>1</Id>
      <Text>Replace</Text>
   </Choices>
   <Choices>
      <Id>2</Id>
      <Text>Cancel</Text>
   </Choices>
</VmPendingQuestion>
```

To supply the answer, POST a `VmQuestionAnswer` element to the `question/action/answer` link of the `Vm`.

Request:

```
POST http://vcloud.example.com/api/vApp/vm-5/question/action/answer
Content-type: application/vnd.vmware.vcloud.vmPendingAnswer+xml
<?xml version="1.0" encoding="UTF-8"?>
<VmQuestionAnswer
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <ChoiceId>2</ChoiceId>
   <QuestionId>50</QuestionId>
</VmQuestionAnswer>
```

Creating and Using vApp Snapshots

You can take a snapshot of all the virtual machines in a vApp. After the snapshots are taken, you can revert all virtual machines in the vApp to the most recent snapshot, or remove all snapshots.

You can take a snapshot of a vApp whether or not it is powered on. The `createSnapshot` operation always creates a snapshot for each virtual machine in the vApp. Snapshots are created in the order specified for virtual machines in the `StartupSection` of the `VApp` element.

vApp snapshots have the following limitations:

- They do not capture the `ovf:ProductSection` elements in the vApp.
- They do not capture NIC configurations.
- They cannot be created if any virtual machine in the vApp is connected to an independent disk.

Prerequisites

- This operation requires the rights included in the predefined **vApp User** role or an equivalent set of rights.
- You must be the owner of the object that this operation affects.

Procedure

1 (Optional) Retrieve information about current vApp snapshots.

When you create a snapshot, it overwrites any existing snapshots without warning. Before creating a new snapshot, you might want to verify whether there are any existing snapshots. Make a request like this one:

```
GET https://vcloud.example.com/api/vApp/vapp-33/snapshotSection
```

The response is a `SnapshotSection` element containing a link to each of the current snapshots for the vApp.

2 Create a `CreateSnapshotParams` request body.

See Create a vApp Snapshot.

3 POST the `CreateSnapshotParams` to the `action/createSnapshot` link of the vApp.

Example: Create a vApp Snapshot

In this example, the `memory` and `quiesce` options are not specified, so their values default to `true`. The `name` attribute is optional. If you omit it from the request, the system generates a name for the snapshot.

Request:

```
POST https://vcloud.example.com/api/vApp/vapp-33/action/createSnapshot
Content-Type: application/vnd.vmware.vcloud.createSnapshotParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CreateSnapshotParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="snap1">
   <Description>Demo snapshot</Description>
</CreateSnapshotParams>
```

The response is a `Task`.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<Task
   ...
   operationName="vappCreateSnapshot"
   ... />
   ...
</Task>
```

After the snapshot is created, you can revert the vApp to the state captured in the snapshot.

```
POST https://vcloud.example.com/api/vApp/vapp-33/action/revertToCurrentSnapshot
```

You can also remove all snapshots.

```
POST https://vcloud.example.com/api/vApp/vapp-33/action/removeAllSnapshots
```

Neither of these requests has a request body. Both return a `Task`.

Attach or Detach a Named Disk

Use the `disk/action/attach` or `disk/action/detach` links in a `Vm` to attach or detach a named disk. You must be the owner of the `Vm` object and the disk.

Every `Vm` element includes links of the following form:

```
<Link
   rel="disk:attach"
   type="application/vnd.vmware.vcloud.diskAttachOrDetachParams+xml"
   href="https://vcloud.example.com/api/vApp/vm-4/disk/action/attach" />
<Link
   rel="disk:detach"
   type="application/vnd.vmware.vcloud.diskAttachOrDetachParams+xml"
   href="https://vcloud.example.com/api/vApp/vm-4/disk/action/detach" />
```

You can POST a `DiskAttachOrDetachParams` element to one of these URLs to attach or detach a named disk.

Deleting a VM of an attached disk in vSphere corrupts the disk state in VMware Cloud Director and you cannot detach or delete the disc. To detach such disks, you can add the `force` parameter to the `disk` API call.

```
https://vcloud.example.com/api/vApp/vm-vm-ID/disk/action/detach?force=true
```

Note You can attach named disks with sharing type `none` to only one VM. You can attach named disks with sharing type `Disk Sharing` or `Controller Sharing` to multiple VMs. For more information about the sharing types, see Creating and Using Named Disks.

Prerequisites

- Verify that you are logged in to the VMware Cloud Director API as an administrator or the object owner.
- Verify that the VDC contains a named disk. If the VDC does not contain named disks, you can create one. See Create an Independent Disk.

Procedure

1. Retrieve a reference to the named disk to attach to a virtual machine.

   The named disk and the virtual machine must be contained by the same VDC. You can retrieve references to named disks in a VDC in the following ways:

   - Use the query service. A query like this one returns references to all `Disk` objects in the VDC named `MyVdc`.

     ```
     GET https://vcloud.example.com/api/query?
     type=disk&format=references&filter=vdcName==MyVdc
     ```

   - Retrieve the VDC and look for `ResourceEntity` elements whose `type` attribute has a value of `application/vnd.vmware.vcloud.disk+xml`. The `href` value of a `ResourceEntity` that represents a disk is a reference to that `Disk` object.

2. Verify that the named disk is not connected to any virtual machine.

3. Create a `DiskAttachOrDetachParams` element.

   Provide a reference to the named disk in the `href` attribute of the `Disk` element.

4. POST the `DiskAttachOrDetachParams` to the `disk/action/attach` link of the `Vm`.

Example: Attach a named Disk to a Virtual Machine

This request attaches the disk created in Create an Independent Disk to a virtual machine.

Request:

```
POST https://vcloud.example.com/api/vApp/vm-4/disk/action/attach
Content-Type: application/vnd.vmware.vcloud.diskAttachOrDetachParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<DiskAttachOrDetachParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Disk
      type="application/vnd.vmware.vcloud.disk+xml"
      href="https://vcloud.example.com/api/disk/128" />
</DiskAttachOrDetachParams>
```

The response is a `Task`.

Response:

```
<Task
   href="https://vcloud.example.com/api/task/57"
   ...
   status="running"
   operationName="vappAttachDisk"
   ... />
   ...
</Task>
```

About Virtual Machine Metrics

Virtual machine metrics provide information about a virtual machine's performance and resource consumption in several categories.

VMware Cloud Director collects metrics for all running virtual machines. You can retrieve virtual machine metrics either as an instantaneous reading taken when the request is made, or as a collection of historic readings taken at regular intervals. Historic metrics are retained for 14 days, and can be retrieved whether or not the virtual machine is powered on.

The following categories of metrics are collected:

cpu

Virtual CPU usage metrics. CPU metrics are an aggregate of all the virtual machine's cores and sockets.

disk

Virtual hard disk capacity and performance metrics. When a virtual machine has multiple disks, metrics are reported as an aggregate for all disks.

mem

Virtual machine memory usage.

For historic and current metrics, you can make a GET request to retrieve all categories of metrics from a virtual machine. A GET request for historic metrics returns only the most recent 24 hours of metric data samples. You can also make a POST request with a body that constrains the response to a subset of current or historic metrics in any or all categories. A POST request for historic metrics can specify a time period of up to 14 days.

Important To support retrieval of historic metrics, VMware Cloud Director must be configured with additional database software dedicated to historic metrics collection. See the *VMware Cloud Director Installation and Upgrade Guide*.

Rolled-Up Metrics for Organizations and VDCs

In a system configured for rolled-up metrics collection, `AdminOrg` and `AdminVDC` elements include links you can use to request cumulative metrics for VMs in all vApps in a VDC or in all VDCs in an organization. See "Install and Configure Optional Database Software to Store and Retrieve Historic Virtual Machine Performance Metrics" in the *VMware Cloud Director Installation, Configuration, and Upgrade Guide*.

Metrics Links in a Vm Element

When a virtual machine is powered on, the `Vm` element that represents it includes four links where the value of the `type` attribute has the form `application/vnd.vmware.vcloud.metrics.*UsageSpec.xml`.

```
<Vm ... >
   ...
   <Link
      rel="down"
      href="https://vcloud.example.com/api/vApp/vm-4/metrics/current"
      type="application/vnd.vmware.vcloud.metrics.currentUsageSpec+xml"/>
   <Link
      rel="down"
      href="https://vcloud.example.com/api/vApp/vm-4/metrics/historic"
      type="application/vnd.vmware.vcloud.metrics.historicUsageSpec+xml"/>
   <Link
      rel="metrics"
      href="https://vcloud.example.com/api/vApp/vm-4/metrics/current"
      type="application/vnd.vmware.vcloud.metrics.currentUsageSpec+xml"/>
   <Link
      rel="metrics"
      href="https://vcloud.example.com/api/vApp/vm-4/metrics/historic"
      type="application/vnd.vmware.vcloud.metrics.historicUsageSpec+xml"/>
   ...
</Vm>
```

- Use the links where `rel="down"` with a GET request to retrieve current or historic metrics in all categories.
- Use the links where `rel="metrics"` with a POST request to retrieve a subset of current or historic metrics.

When a virtual machine is powered off, you cannot retrieve current metrics from it, so the `.../metrics/current` links are not returned in the `Vm` element.

Negative Values for Current Metrics

Current metrics can be returned with negative values if the specific metric being requested does not have a valid current value. This condition can exist if you make a metrics request before the first collection interval for that metric elapses. For example, a request for CPU usage metrics made shortly after a virtual machine is powered on can report negative values.

Retrieving All Current or Historic Metrics

You can make a GET request to either of the `metrics` links in a `Vm` object to retrieve all current or historic metrics.

Current metrics are instantaneous values collected at the time the request is processed by the server. A GET request for current metrics retrieves a list of current metric values, one for each category, and returns it as a single XML document. You can retrieve a subset of current metrics by making a POST request with a `CurrentUsageSpec` body.

Historic metrics are collected and stored for 14 days. A GET request for historic metrics retrieves the past 24 hours of metric history and returns it as a single XML document. In the typical case, this document contains thousands of lines. You can retrieve a subset of historic metrics by making a POST request with a `HistoricUsageSpec` body.

Prerequisites

This operation requires the rights included in the predefined **vApp User** role or an equivalent set of rights.

See "Install and Configure Optional Database Software to Store and Retrieve Historic Virtual Machine Performance Metrics" in the *VMware Cloud Director Installation, Configuration, and Upgrade Guide* for information about configuring the system to collect rolled-up metrics for VDCs and organisations.

Procedure

1 Retrieve the XML representation of the virtual machine.

Use a request like this one:

```
GET https://vcloud.example.com/api/vApp/vm-4
```

2 Examine the `Vm` element to find the metrics links that it contains.

These links have the following form, where *id* is the virtual machine's unique identifier:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.metrics.currentUsageSpec+xml"
   href="https://vcloud.example.com/api/vApp/vm-id/metrics/current"/>
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.metrics.historicUsageSpec+xml"
   href="https://vcloud.example.com/api/vApp/vm-id/metrics/historic"/>
```

3 Make a GET request to the appropriate link.

Option	Request
Retrieve all current metrics	GET https://vcloud.example.com/api/vApp/vm-*id/*metrics/current
Retrieve all historic metrics for the past 24 hours	GET https://vcloud.example.com/api/vApp/vm-*id/*metrics/historic

See Retrieving All Current or Historic Metrics

Example: Retrieving All Current or Historic Metrics

This request retrieves all current metrics from a virtual machine.

Request

```
GET https://vcloud.example.com/api/vApp/vm-4/metrics/current
```

Response

```
200 OK
Content-Type: application/vnd.vmware.vcloud.metrics.currentusage+xml
...
<CurrentUsage
   xmlns="http://www.vmware.com/vcloud/v1.5"
   ... >
   <Link
      rel="up"
      href="https://vcloud.example.com/api/vApp/vm-4"
      type="application/vnd.vmware.vcloud.vm+xml"/>
   <Metric
      name="cpu.usage.average"
      unit="PERCENT"
      value="33.7"/>
   <Metric
      name="cpu.usage.maximum"
      unit="PERCENT"
      value="89.0"/>
   <Metric
      name="cpu.usagemhz.average"
      unit="MEGAHERTZ"
      value="262.0"/>
   <Metric
      name="mem.usage.average"
      unit="PERCENT"
      value="44.0"/>
   <Metric
      name="disk.provisioned.latest"
      unit="KILOBYTE"
      value="2190418.0"/>
   <Metric
      name="disk.write.average"
      unit="KILOBYTES_PER_SECOND"
      value="20.0"/>
   <Metric
      name="disk.used.latest"
      unit="KILOBYTE"
      value="1633362.0"/>
   <Metric
      name="disk.read.average.0"
      unit="KILOBYTES_PER_SECOND"
      value="44.0"/>
   <Metric
      name="disk.read.average"
      unit="KILOBYTES_PER_SECOND"
      value="44.0"/>
</CurrentUsage>
```

This request retrieves the past 24 hours of metrics from a virtual machine. The typical response contains thousands of lines, so only a subset appears here.

Request

```
GET https://vcloud.example.com/api/vApp/vm-4/metrics/historic
```

Response

```
200 OK
Content-Type: application/vnd.vmware.vcloud.metrics.historicusage+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<HistoricUsage xmlns="http://www.vmware.com/vcloud/v1.5">
   <Link
      rel="up"
      href="https://vcloud.example.com/api/vApp/vm-4"
      type="application/vnd.vmware.vcloud.vm+xml"/>
   <MetricSeries
      expectedInterval="1800"
      name="disk.provisioned.latest"
      unit="KILOBYTE">
      <Sample
         timestamp="2013-12-02T20:00:00.000Z"
         value="875295.0"/>
      <Sample
         timestamp="2013-12-02T20:30:00.000Z"
         value="741388.0"/>
   </MetricSeries>
   <MetricSeries
      expectedInterval="300"
      name="cpu.usagemhz.average"
      unit="MEGAHERTZ">
      <Sample
         timestamp="2013-12-02T20:03:20.000Z"
         value="505.0"/>
      <Sample
         timestamp="2013-12-02T20:06:40.000Z"
         value="122.93333333333334"/>
      <Sample ... />
   </MetricSeries>
   <MetricSeries
      expectedInterval="300"
      name="disk.read.average"
      unit="KILOBYTES_PER_SECOND">
      <Sample
         timestamp="2013-12-02T20:03:20.000Z"
         value="208.7"/>
      <Sample
         timestamp="2013-12-02T20:06:40.000Z"
         value="0.0"/>
      <Sample ... />
   </MetricSeries>
   <MetricSeries
      expectedInterval="300"
      name="mem.usage.average"
      unit="PERCENT">
      <Sample
```

```
         timestamp="2013-12-02T20:03:00.000Z"
         value="55.26272895119407"/>
      <Sample
         timestamp="2013-12-02T20:06:40.000Z"
         value="47.19066823323568"/>
      <Sample ... />
   </MetricSeries>
   <MetricSeries
      expectedInterval="300"
      name="cpu.usage.average"
      unit="PERCENT">
      <Sample
         timestamp="2013-12-02T20:03:20.000Z"
         value="24.07100028991699"/>
      <Sample
         timestamp="2013-12-02T20:06:40.000Z"
         value="5.87066666285197"/>
      <Sample ... />
   </MetricSeries>
   <MetricSeries
      expectedInterval="1800"
      name="disk.used.latest"
      unit="KILOBYTE">
      <Sample
         timestamp="2013-12-02T20:00:00.000Z"
         value="262154.0"/>
      <Sample
         timestamp="2013-12-02T20:30:00.000Z"
         value="373779.0"/>
   </MetricSeries>
   <MetricSeries
      expectedInterval="300"
      name="disk.write.average"
      unit="KILOBYTES_PER_SECOND">
      <Sample
         timestamp="2013-12-02T20:03:20.000Z"
         value="30.3"/>
      <Sample
         timestamp="2013-12-02T20:06:40.000Z"
         value="0.5333333333333333"/>
      <Sample ... />
   </MetricSeries>
   <MetricSeries
      expectedInterval="300"
      name="cpu.usage.maximum"
      unit="PERCENT">
      <Sample
         timestamp="2013-12-02T20:03:20.000Z"
         value="24.07100028991699"/>
      <Sample
         timestamp="2013-12-02T20:06:40.000Z"
         value="5.87066666285197"/>
      <Sample ... />
   </MetricSeries>
</HistoricUsage>
```

Note In a system configured for rolled-up metrics collection, `Org` and `Vdc` elements include links you can use to retrieve cumulative metrics for VMs in all vApps in a VDC or in all VDCs in an organization. Requests to these links take the same form as requests shown in these examples, but return the total, for all VMs in all vApps in the container, for each metric subset.

Retrieving Metrics Subsets

To retrieve a subset of current or historic metrics, make a POST request to a current or historic metrics URL. The request body specifies the metrics you want.

The VMware Cloud Director API defines two request bodies, `CurrentUsageSpec` and `HistoricUsageSpec`, that you can use to constrain a metrics request to a subset of available metrics. With these request bodies, you can specify metrics by name, or supply a pattern that uses a wildcard to match a set of names. You can also specify a collection interval for historic metrics. See Metric Names and Patterns.

Prerequisites

This operation requires the rights included in the predefined **vApp User** role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the virtual machine.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/vApp/vm-4
   ```

2 Examine the `Vm` element to find the metrics links where `rel="metrics"`.

   These links have the following form, where *id* is the virtual machine's unique identifier:

   ```
   <Link
      rel="metrics"
      type="application/vnd.vmware.vcloud.metrics.currentUsageSpec+xml"
      href="https://vcloud.example.com/api/vApp/vm-id/metrics/current"/>
   <Link
      rel="metrics"
      type="application/vnd.vmware.vcloud.metrics.historicUsageSpec+xml"
      href="https://vcloud.example.com/api/vApp/vm-id/metrics/historic"/>
   ```

3 Create the appropriate request body.

The request body specifies the metrics you want.

Option	Request
Retrieve a subset of current metrics	Create a `CurrentUsageSpec` element to use as the request body.
Retrieve a subset of historic metrics	Create a `HistoricUsageSpec` element to use as the request body.

4 POST the request body to the appropriate metrics link.

See Retrieving a Subset of Current or Historic Metrics

Note If you do not include a request body, the request is processed as though you had made a GET request.

Example: Retrieving a Subset of Current or Historic Metrics

This example uses a POST request with a `CurrentUsageSpec` body to request a subset of the current metrics retrieved in Retrieving All Current or Historic Metrics. This example also shows the use of wildcards in a `MetricPattern` element. See Metric Patterns.

Request

```
POST https://vcloud.example.com/api/vApp/vm-4/metrics/current
Content-type: application/vnd.vmware.vcloud.metrics.currentUsageSpec+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CurrentUsageSpec xmlns="http://www.vmware.com/vcloud/v1.5">
   <MetricPattern>*.average</MetricPattern>
</CurrentUsageSpec>
```

Response

```
200 OK
Content-Type: application/vnd.vmware.vcloud.metrics.currentusage+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CurrentUsage
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Link
      rel="up"
      href="https://vcloud.example.com/api/vApp/vm-4f"
      type="application/vnd.vmware.vcloud.vm+xml"/>
   <Metric
      name="cpu.usage.average"
      unit="PERCENT"
      value="5.36"/>
   <Metric
      name="cpu.usagemhz.average"
      unit="MEGAHERTZ"
      value="112.0"/>
   <Metric
      name="mem.usage.average"
      unit="PERCENT"
      value="2.99"/>
   <Metric
      name="disk.write.average"
      unit="KILOBYTES_PER_SECOND"
      value="0.0"/>
   <Metric
      name="disk.read.average"
      unit="KILOBYTES_PER_SECOND"
      value="0.0"/>
</CurrentUsage>
```

This example uses a POST request with a `HistoricUsageSpec` body to request a subset of the historic metrics retrieved in Retrieving All Current or Historic Metrics. This example also shows the use of wildcards and a time specification for the past 8 hours.

Request

```
POST https://vcloud.example.com/api/vApp/vm-4/metrics/historic
Content-type: application/vnd.vmware.vcloud.metrics.historicUsageSpec+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<HistoricUsageSpec xmlns="http://www.vmware.com/vcloud/v1.5">
   <RelativeStartTime
      interval="8"
      unit="hour"/>
   <RelativeEndTime
      interval="0"
      unit="hour"/>
   <MetricPattern>cpu.*</MetricPattern>
   <MetricPattern>disk.*</MetricPattern>
</HistoricUsageSpec>
```

The full response contains several thousand lines, so only a subset appears here.

Response

```
200 OK
Content-Type: application/vnd.vmware.vcloud.metrics.historicusage+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<HistoricUsage
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Link
      rel="up"
      href="https://vcloud.example.com/api/vApp/vm-4"
      type="application/vnd.vmware.vcloud.vm+xml"/>
   <MetricSeries
      expectedInterval="1800"
      name="disk.provisioned.latest"
      unit="KILOBYTE">
      <Sample
         timestamp="2013-12-02T20:00:00.000Z"
         value="875295.0"/>
```

```
       <Sample
          timestamp="2013-12-02T20:30:00.000Z"
          value="741388.0"/>
    </MetricSeries>
    <MetricSeries
       expectedInterval="300"
       name="cpu.usagemhz.average"
       unit="MEGAHERTZ">
       <Sample
          timestamp="2013-12-02T20:03:20.000Z"
          value="505.0"/>
       <Sample
          timestamp="2013-12-02T20:06:40.000Z"
          value="122.93333333333334"/>
       <Sample/>
    </MetricSeries>
    <MetricSeries
       expectedInterval="300"
       name="disk.read.average"
       unit="KILOBYTES_PER_SECOND">
       <Sample
          timestamp="2013-12-02T20:03:20.000Z"
          value="208.7"/>
       <Sample
          timestamp="2013-12-02T20:06:40.000Z"
          value="0.0"/>
        <Sample ... />
    </MetricSeries>
    <MetricSeries
       expectedInterval="300"
       name="mem.usage.average"
       unit="PERCENT">
       <Sample
          timestamp="2013-12-02T20:03:00.000Z"
          value="55.26272895119407"/>
       <Sample
          timestamp="2013-12-02T20:06:40.000Z"
          value="47.19066823323568"/>
       <Sample ... />
    </MetricSeries>
    <MetricSeries
       expectedInterval="300"
       name="cpu.usage.average"
       unit="PERCENT">
       <Sample
          timestamp="2013-12-02T20:03:20.000Z"
          value="24.07100028991699"/>
       <Sample
          timestamp="2013-12-02T20:06:40.000Z"
          value="5.87066666285197"/>
       <Sample/>
    </MetricSeries>
    <MetricSeries
       expectedInterval="1800"
       name="disk.used.latest"
```

```
      unit="KILOBYTE">
      <Sample
         timestamp="2013-12-02T20:00:00.000Z"
         value="262154.0"/>
      <Sample
         timestamp="2013-12-02T20:30:00.000Z"
         value="373779.0"/>
   </MetricSeries>
   <MetricSeries
      expectedInterval="300"
      name="disk.write.average"
      unit="KILOBYTES_PER_SECOND">
      <Sample
         timestamp="2013-12-02T20:03:20.000Z"
         value="30.3"/>
      <Sample
         timestamp="2013-12-02T20:06:40.000Z"
         value="0.5333333333333333"/>
       <Sample... />
   </MetricSeries>
   <MetricSeries
      expectedInterval="300"
      name="cpu.usage.maximum"
      unit="PERCENT">
      <Sample
         timestamp="2013-12-02T20:03:20.000Z"
         value="24.07100028991699"/>
      <Sample
         timestamp="2013-12-02T20:06:40.000Z"
         value="5.87066666285197"/>
      <Sample ... />
   </MetricSeries>
</HistoricUsage>
```

Metric Names and Patterns

When you retrieve a subset of current or historic metrics, you can specify the metrics by name, or by using a pattern that includes a wildcard character.

Metric Names

Metric names are dot-separated strings.

Table 5-4. Metric Names

Metric Name	Type	Unit	Description
`cpu.usage.average`	rate	percent	Host view of this virtual machine's average actively used CPU as a percentage of total available. Includes all cores in all sockets.
`cpu.usagemhz.average`	rate	megahertz	Host view of this virtual machine's average actively used CPU as a raw measurement . Includes all cores in all sockets.
`cpu.usage.maximum`	rate	percent	Host view of this virtual machine's maximum actively used CPU as a percentage of total available. Includes all cores in all sockets.
`mem.usage.average`	absolute	percent	Memory used by this virtual machine as a percentage of total configured memory.
`disk.provisioned.latest`	absolute	kilobytes	Storage space allocated to this virtual hard disk in the containing organization virtual data center.
`disk.used.latest`	absolute	kilobytes	Storage used by all virtual hard disks.
`disk.read.average`	rate	kilobytes per second	Average read rate for all virtual hard disks.
`disk.write.average`	rate	kilobytes per second	Average write rate for all virtual hard disks.

Note When a virtual machine has multiple disks, metrics are reported as an aggregate for all disks. CPU metrics are an aggregate of all cores and sockets.

Metric Patterns

A `CurrentUsageSpec` or `HistoricUsageSpec` can include `MetricPattern` elements that specify multiple metric names using a wildcard character. To form a metric pattern, replace any component of a metric name with an asterisk, which is a wildcard that matches all values of the metric name component it replaces. For example, this `MetricPattern` matches all metric names that begin with `disk`.

```
<MetricPattern>disk.*</MetricPattern>
```

The response would include these metric names: `disk.provisioned.latest`, `disk.used.latest`, `disk.read.average`, and `disk.write.average`.

A different `MetricPattern` matches all metric names that begin with `disk` and end with `average`.

```
<MetricPattern>disk.*.average</MetricPattern>
```

The response would include the metric names `disk.read.average` and `disk.write.average`.

Metric Series Expected Intervals and Timestamps

A `HistoricUsage` element includes zero or more `MetricSeries` elements, each of which includes a set of `Sample` elements. Each `MetricSeries` has an `expectedInterval` attribute that specifies the interval, in milliseconds, at which the samples in the series are reported. Each `Sample` in the `MetricSeries` has a `timestamp` attribute noting the absolute time at which the sample was taken. You can use the `timestamp` and `expectedInterval` values to aggregate sample data, and to determine when metrics became unavailable because the virtual machine was powered off or unreachable.

Specifying Collection Start and End Times

A `HistoricUsageSpec` can include a time specification that constrains the result set to metrics collected between a start time and an end time. This time specification can be relative or absolute.

In `RelativeStartTime` and `RelativeEndTime` elements, start and end times are specified as an interval and a unit, which are interpreted as `interval unit`s ago. For example, this `HistoricUsageSpec` requests metrics collected during the past 8 hours.

```
<HistoricUsageSpec xmlns="http://www.vmware.com/vcloud/v1.5">
   <RelativeStartTime
      interval="8"
      unit="hour"/>
   <RelativeEndTime
      interval="0"
      unit="hour"/>
   ...
</HistoricUsageSpec>
```

You can also write this specification with no `RelativeEndTime` element, rather than a `RelativeEndTime` with an `interval` attribute value of `0`. Both constructions specify an end time of now.

Alternatively, you can use `AbsoluteStartTime` and `AbsoluteEndTime` elements to specify absolute start and end times in a `HistoricUsageSpec`, as shown in this example, which returns metrics recorded during a one hour period:

```
<HistoricUsageSpec xmlns="http://www.vmware.com/vcloud/v1.5">
   <AbsoluteStartTime time="2013-11-13T10:00:00.000Z" />
   <AbsoluteEndTime time="2013-11-13T11:00:00.000Z" />
   ...
</HistoricUsageSpec>
```

Create a Virtual Machine with Custom Resource Allocation Settings

By default, new and imported virtual machines obtain their resource allocation settings from the organization virtual data center on which they reside. System administrators can create a virtual machine with custom CPU and memory reservation, limit, and shares settings that are different from the values configured for the target organization VDC.

Typically, when creating virtual machines, users cannot configure custom VM reservation, limit, and shares settings. Importing or adopting a virtual machine does not retain the VM reservation, limit, and shares settings that are configured in vCenter Server. Uploading OVF templates does not retain the VM reservation, limit, and shares settings too. Newly created or imported virtual machines obtain their resource allocation settings from their organization virtual data centers.

During an instantiation from a template, you cannot reconfigure the VM reservation, limit, and shares settings.

By using the `VmSpecSection` section, you can create a VM with custom reservation, limit, and shares values for both CPU and memory. You can also use this section to reconfigure the resource allocation settings of a virtual machine.

Note The custom VM resource allocation settings must not exceed the default values per VM of the organization VDC.

When you upload a VM with custom reservation, limit, and shares settings as a VM template to a catalog, the VM template retains the custom settings. Instantiations from a VM template with custom reservation, limit, and shares settings retain the custom settings but might fail if the custom settings exceed the default values per VM for the target VDC.

Copying or moving a VM with custom reservation, limit, and shares settings retains the custom settings but might fail if the custom settings exceed the default values per VM for the target VDC.

Prerequisites

This operation is restricted to system administrators and users assigned with roles that have the right `vApp: Edit VM CPU and Memory Reservation / Limit / Shares in all VDC types`.

Procedure

1 Retrieve the XML representation of the organization VDC on which you want to create the new virtual machine.

  Use a request similar to:

```
GET https://vcloud.example.com/api/vdc/23
```

The response contains a `Link` element for creating a standalone VM similar to:

```
<Link
 href="https://vcloud.example.com/api/vdc/23/action/createvm"
 rel="add"
 type="application/vnd.vmware.vcloud.CreateVmParams+xml"/>
```

2 Create a `CreateVmParams` element with a `VmSpecSection` section that includes the custom reservation, limit, and shares settings for the new VM.

3 POST the `CreateVmParams` element that you created in Step 2 to the URL from the response in Step 1.

See the request portion of Create a VM with Custom CPU and Memory Reservation, Limit, and Shares Settings.

Example: Create a VM with Custom CPU and Memory Reservation, Limit, and Shares Settings

Request:

```
POST https://vcloud.example.com/api/vdc/23/action/createvm
Content-Type: application/vnd.vmware.vcloud.CreateVmParams+xml
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CreateVmParams
   xmlns:ns3="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
   xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
   xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
   xmlns:ns6="http://www.vmware.com/schema/ovf"
   xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1"
   xmlns:ns8="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:ns9="http://www.vmware.com/vcloud/versions">
   <Description>some description</Description>
   <CreateVm
      name="blank-vm1-name">
      <VmSpecSection Modified="true">
         <ns2:Info>Virtual Machine specification</ns2:Info>
         <OsType>windows9Server64Guest</OsType>
         <NumCpus>1</NumCpus>
         <NumCoresPerSocket>1</NumCoresPerSocket>
         <CpuResourceMhz>
            <Configured>1</Configured>
            <Reservation>0</Reservation>
            <Limit>1000</Limit>
            <SharesLevel>CUSTOM</SharesLevel>
            <Shares>1000</Shares>
         </CpuResourceMhz>
         <MemoryResourceMb>
            <Configured>1000</Configured>
            <Reservation>600</Reservation>
            <Limit>1000</Limit>
            <SharesLevel>CUSTOM</SharesLevel>
            <Shares>1000</Shares>
         </MemoryResourceMb>
         <DiskSection>
            <DiskSettings>
               <SizeMb>4</SizeMb>
               <UnitNumber>0</UnitNumber>
               <BusNumber>0</BusNumber>
               <AdapterType>4</AdapterType>
               <ThinProvisioned>false</ThinProvisioned>
               <StorageProfile
                  href="https://vcloud.example.com/api/vdcStorageProfile/71"
                  id="71"
                  name="*"
                  type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"/>
               <overrideVmDefault>true</overrideVmDefault>
            </DiskSettings>
         </DiskSection>
         <HardwareVersion>vmx-11</HardwareVersion>
         <VirtualCpuType>VM64</VirtualCpuType>
      </VmSpecSection>
      <GuestCustomizationSection>
         <ns2:Info>Guest Customization specification</ns2:Info>
         <ComputerName>computername</ComputerName>
      </GuestCustomizationSection>
   </CreateVm>
</CreateVmParams>
```

Response:

```
Response-Code: 202
Content-type: application/vnd.vmware.vcloud.task+xml
...

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Task
   ...
   operation="Composing blank-vm1-name-cb (86)"
   operationName="vdcComposeVapp"
   ...>
   <Owner
      .../>
   <User
      .../>
  <Organization
      .../>
  <Progress>1</Progress>
  <Details/>
</Task>
```

Reconfiguring vApps and Virtual Machines

6

You establish the initial configuration of a vApp and its virtual machines when you create it. After a vApp has been created by instantiation, cloning, composition, or recomposition, you can make further changes to its configuration using PUT requests that update modifiable elements of vApp and virtual machine objects.

Nearly all of the properties that you can specify when you create a vApp using any of the requests listed in Instantiation Parameters can be modified using a common reconfiguration workflow. Modifiable elements include a `Link` element where `rel="edit"`. See Retrieve the Configuration Links for a vApp and Retrieve the Configuration Links for a Virtual Machine.

Before you create or update the virtual hardware configuration of a VM, retrieve the list of virtual hardware versions supported by the organization VDC where the VM will be deployed. See Retrieve a List of Supported Virtual Hardware Versions and Guest Operating Systems.

Reconfiguration Workflow

The workflow for reconfiguring a vApp or virtual machine is the same regardless of the section you are modifying.

1 Retrieve the `vApp` or `Vm` and examine the response to find the section that you want to modify.

2 Retrieve the section by making a GET request to the URL in the section's `href` attribute value.

3 Modify the section as needed.

4 Update the section by making a PUT request to the section's edit link, a `Link` element in the section where `rel="edit"`, and supplying the modified section in the request body.

5 To preserve these modification, you can capture the reconfigured vApp to create a new vApp template. See Capture a vApp as a Template.

Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

Note You cannot make configuration changes to a vApp if it is in maintenance mode. A system administrator can put a vApp into maintenance mode to prevent metadata changes during administrative operations such as backup, restore, and upgrade.

For a full list of the VMware Cloud Director API user operations, see the *VMware Cloud API Schema Reference* at https://code.vmware.com.

Read the following topics next:

- Retrieve the Configuration Links for a vApp
- Retrieve the Configuration Links for a Virtual Machine
- Retrieve a List of Supported Virtual Hardware Versions and Guest Operating Systems
- Update Multiple Sections of a Virtual Machine
- Retrieve or Update a Modifiable Section
- Update a vApp Network Configuration
- Update the NetworkConnectionSection of a Virtual Machine
- Retrieve or Modify the CPU Configuration of a Virtual Machine
- Retrieve or Modify the GuestCustomizationSection of a Virtual Machine
- Retrieve or Modify ProductSection Elements
- Retrieve or Modify Groups of Related Sections in a Virtual Machine
- Retrieve or Modify the Hard Disk Configuration of a Virtual Machine
- Retrieve or Update the VDC Compute Policy of a Virtual Machine
- Update the Storage Profile for a Virtual Machine
- Override the Default Storage Profile for a Hard Disk
- Specify Hard Disk IOPS

Retrieve the Configuration Links for a vApp

Each modifiable section of a vApp includes a `Link` element whose `rel` attribute has the value `edit`. You cannot modify sections that do not contain this `Link` element.

Any `ovf:SectionType` element can include an arbitrary number of `Link` elements. Sections that you can modify include a `Link` element where `rel="edit"`. To modify one of these sections, retrieve it by making a GET request to the URL in the section's `href` attribute. Then make a PUT request to the `href` attribute value of the `Link` where `rel="edit"` to update the section with your modifications.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the vApp.

   Use a GET request as shown in Configuration Links in a vApp.

2 Examine the response for `edit` links to modifiable sections.

   The response portion of Configuration Links in a vApp includes one of these links for each of the modifiable sections of the vApp. You cannot modify sections that do not contain a `Link` element where `rel="edit"`.

Example: Configuration Links in a vApp

In this example, the response was edited to show only the modifiable sections of the `VApp` element. Each `Vm` in the `Children` element of the `VApp` includes additional configuration links, shown in Configuration Links in a Vm Element.

Request:

```
GET https://vcloud.example.com/api/vApp/vapp-7
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp ... href="https://vcloud.example.com/api/vApp/vapp-7">
   ...
   <LeaseSettingsSection ...
      href="https://vcloud.example.com/api/vApp/vapp-7/leaseSettingsSection/" ...>
      ...
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.leaseSettingsSection+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/leaseSettingsSection/" />
      ...
   </LeaseSettingsSection>
   <ovf:StartupSection ...
      href="https://vcloud.example.com/api/vApp/vapp-7/startupSection/" ... >
      ...
      <Link
```

```
         rel="edit"
         type="application/vnd.vmware.vcloud.startupSection+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/startupSection/" />
   ...
   </ovf:StartupSection>
   <NetworkConfigSection ...
      href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/" ... />
      ...
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.networkConfigSection+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/" />
      ...
   </NetworkConfigSection>
   <Children>
      <Vm ...
         status="8"
         name="ubuntu10-x86"
         href="https://vcloud.example.com/api/vApp/vm-4">
         ...
      </Vm>
   </Children>
</VApp>
```

Retrieve the Configuration Links for a Virtual Machine

A virtual machine is represented by a `Vm` element. Each modifiable section of a `Vm` element includes a `Link` element whose `rel` attribute has the value `edit`. You cannot modify sections that do not contain this `Link` element.

Any `ovf:SectionType` element can include an arbitrary number of `Link` elements. Sections that you can modify include a `Link` element where `rel="edit"`. To modify one of these sections, retrieve it by making a GET request to the URL in the section's `href` attribute. Then make a PUT request to the `href` attribute value of the `Link` where `rel="edit"` to update the section with your modifications.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the vApp that contains the virtual machine to reconfigure.

  Use a GET request as shown in Configuration Links in a vApp.

2 In the `VApp` element's `Children` element, find the `Vm` element that represents the virtual machine and retrieve it.

3 Examine the response to find the `reconfigureVm` link and `edit` links to modifiable sections.

The response portion of Configuration Links in a Vm Element shows the `reconfigureVm` link and links for each of the modifiable sections of the `Vm`. You cannot modify sections that do not contain a link where `rel="edit"`.

Example: Configuration Links in a Vm Element

This example retrieves a `Vm` element shown in Configuration Links in a vApp. It expands that element to show its configuration links. It also shows the entire `NetworkConnectionSection` of that `Vm`, and additional information that is referenced in other examples. You cannot modify sections that do not have a `Link` where `rel="edit"`, so they do not appear in this example. Modifiable sections of the parent vApp are shown in Configuration Links in a vApp.

Request:

```
GET https://vcloud.example.com/api/vApp/vm-4
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.vm+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Vm
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   status="8"
   name="ubuntu10-x86"
   href="https://vcloud.example.com/api/vApp/vm-4">
   ...
   <Link
      rel="reconfigureVm"
      type="application/vnd.vmware.vcloud.vm+xml"
      name="vm-4"
      href="https://vcloud.example.com/api/vApp/vm-4/action/reconfigureVm" />
   ...
   <ovf:VirtualHardwareSection>
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.virtualHardwareSection+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/" />
      <Link
         rel="down"
         type="application/vnd.vmware.vcloud.rasdItem+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu" />
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.rasdItem+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu" />
      <Link
         rel="down"
         type="application/vnd.vmware.vcloud.rasdItem+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/memory" />
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.rasdItem+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/memory" />
      <Link
         rel="down"
         type="application/vnd.vmware.vcloud.rasdItemsList+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks" />
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.rasdItemsList+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks" />
      <Link
         rel="down"
         type="application/vnd.vmware.vcloud.rasdItemsList+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/media" />
      <Link
         rel="down"
         type="application/vnd.vmware.vcloud.rasdItemsList+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/
networkCards" />
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.rasdItemsList+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/
networkCards" />
   </ovf:VirtualHardwareSection>
   <ovf:OperatingSystemSection
      href="https://vcloud.example.com/api/vApp/vm-4/operatingSystemSection"
      ovf:id="1">
      <ovf:Info>Specifies the operating system installed
      </ovf:Info>
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.operatingSystemSection+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/operatingSystemSection" />
   </ovf:OperatingSystemSection>
   <NetworkConnectionSection>
      <ovf:Info>Specifies the available VM network connections</ovf:Info>
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
         href="https://vcloud.example.com/api/vApp/vm-4/networkConnectionSection" />
      ...
   </NetworkConnectionSection>
   <GuestCustomizationSection>
      <Link
         rel="edit"
         href="https://vcloud.example.com/api/vApp/vm-4/guestCustomizationSection+xml"
         type="application/vnd.vmware.vcloud.guestCustomizationSection+xml">
      </Link>
      ...
   </GuestCustomizationSection>
   <VmCapabilities>
```

```
      <Link
         rel="edit"
         href="https://vcloud.example.com/api/vApp/vm-4/vmCapabilities/"
         type="application/vnd.vmware.vcloud.vmCapabilitiesSection+xml" />
      <MemoryHotAddEnabled>false</MemoryHotAddEnabled>
      <CpuHotAddEnabled>false</CpuHotAddEnabled>
   </VmCapabilities>
   <BootOptions>
      <Link
         rel="edit"
         href="https://vcloud.example.com/api/vApp/vm-4/action/bootOptions"
         type="application/vnd.vmware.vcloud.bootOptionsSection+xml" />
      <BootDelay>0</BootDelay>
      <EnterBIOSSetup>false</EnterBIOSSetup>
   </BootOptions>
   ...
</Vm>
```

VMware Cloud Director API Custom Attributes

VMware Cloud Director API custom attributes extend several elements in the `ovf` and `rasd` XML namespaces. You can use these attributes to provide additional detail about virtual NIC and hard disk controller devices, or to specify the guest operating system type.

With the exception of `osType`, custom attributes are scoped to `ovf:Item` elements based on the elements' RASD resource type. The `osType` attribute applies to the `ovf:OperatingSystemSection` element. All of the elements to which these custom attributes apply are contained in the `VirtualHardwareSection` of a `Vm`.

Table 6-1. VMware Cloud Director API Custom Attributes for OVF and RASD Elements

Element Name	RASD Resource Type	Attribute Name	Attribute Type	Description
rasd:Connection	10 (Network adapters)	ipAddressingMode	xs:string	IP addressing mode to use for this connection. One of `NONE`, `MANUAL`, `DHCP`, `POOL`.
rasd:Connection	10 (Network adapters)	ipAddress	xs:string	If `ipAddressingMode="MANUAL"`, set the IP address here
rasd:Connection	10 (Network adapters)	primaryNetworkConnection	xs:boolean	True if this is the primary network connection of the virtual machine
rasd:HostResource	17 (Hard disks)	capacity	xs:string	Hard disk capacity in megabytes. See Retrieve or Modify the Hard Disk Configuration of a Virtual Machine
rasd:HostResource	17 (Hard disks)	busType	xs:string	Bus type. One of: ■ 5 (IDE) ■ 6 (SCSI) ■ 20 (SATA)

Table 6-1. VMware Cloud Director API Custom Attributes for OVF and RASD Elements (continued)

Element Name	RASD Resource Type	Attribute Name	Attribute Type	Description
rasd:HostResource	17 (Hard disks)	busSubType	xs:string	Hard disk controller type. One of: - buslogic - lsilogic - lsilogicsas - VirtualSCSI - vmware.sata.ahci See Retrieve or Modify the Hard Disk Configuration of a Virtual Machine for valid combinations of `busType` and `busSubType`.
rasd:HostResource	17 (Hard disks)	storageProfileOverrideVmDefault	xs:boolean	If `true`, the storage profile specified in `storageProfileHref` is always used for this disk regardless of the value specified in the virtual machine's `StorageProfile`. . If `false`, any storage profile is specified in `storageProfileHref` is ignored and the disk is migrated to the storage specified in the virtual machine's `StorageProfile`.
rasd:HostResource	17 (Hard disks)	storageProfileHref	xs:string	If `storageProfileOverrideVmDefault` is `true`, the value of `storageProfileHref` specifies the storage profile to use for this hard disk. The storage profile must be available in the VDC where this virtual machine is deployed.
rasd:HostResource	17 (Hard disks)	disk	xs:string	Read-only reference to an attached independent disk.
rasd:HostResource	17 (Hard disks)	iops	xs:int	Requested I/O operations per second for this hard disk. See Specify Hard Disk IOPS.
ovf:OperatingSystemSection	N/A	osType	xs:string	Internal VMware identifier for the guest operating system. See Enum - VirtualMachineGuestOsIdentifier in the VMware vSphere API Reference documentation.

For more information about OVF and RASD (`CIM_ResourceAllocationSettingData`) elements, see About OVF and About DMTF, CIM, and RASD.

Retrieve a List of Supported Virtual Hardware Versions and Guest Operating Systems

Before you update the virtual hardware configuration or choose a guest operating system for a VM, retrieve the list of virtual hardware versions supported by the organization VDC where the VM will be deployed. The virtual hardware version and guest operating system that you choose for a VM constrain many of the configuration values that you can specify. In most cases, you should choose the highest virtual hardware version supported by the VDC.

The set of virtual hardware versions supported by an organization VDC is contained in its `Capabilities` element. You can retrieve any of the supported virtual hardware versions to see a comprehensive list of guest operating systems it supports and, for each guest operating system, a list of the virtual hardware that it supports.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the VDC and look for the `Capabilities` element it contains.

This element contains a `SupportedHardwareVersions` element that lists the virtual hardware versions that the VDC supports. The highest supported virtual hardware version is the default.

```
<Vdc ...>
   ...
   <Capabilities>
      <SupportedHardwareVersions>
         <SupportedHardwareVersion
            href="https://vcloud.example.com/api/hwv/vmx-09"
            name="vmx-09"
            type="application/vnd.vmware.vcloud.virtualHardwareVersion+xml">vmx-09</
SupportedHardwareVersion>
         <SupportedHardwareVersion
            href="https://vcloud.example.com/api/hwv/vmx-10"
            name="vmx-10"
            type="application/vnd.vmware.vcloud.virtualHardwareVersion+xml">vmx-10</
SupportedHardwareVersion>
         <SupportedHardwareVersion
            default="true"
            href="https://vcloud.example.com/api/hwv/vmx-11"
            name="vmx-11"
            type="application/vnd.vmware.vcloud.virtualHardwareVersion+xml">vmx-11</
SupportedHardwareVersion>
```

```
      </SupportedHardwareVersions>
   </Capabilities>
   ...
</Vdc>
```

2 Retrieve a `SupportedHardwareVersion` from the list.

   This element contains a `VirtualHardwareVersion` element that lists the guest operating systems supported by that hardware version.

3 Retrieve an `OperatingSystem` from the list.

   Operating systems are grouped into families, each of which is contained in an `OperatingSystemFamilyInfo` element. Each `OperatingSystem` element provides information about supported virtual hardware elements such as hard disks, NICs, and CPUs, as well as guest customization and personalization, CIM and OVF identifiers, Microsoft sysprep capabilities, and so on. See Virtual Hardware Version Details for an Operating System.

Example: Virtual Hardware Version Details for an Operating System

This example retrieves the default virtual hardware version from the list shown in Step 1.

Request:

```
GET https://vcloud.example.com/api/hwv/vmx-11
```

Response:

```
200 OK
...
<VirtualHardwareVersion ... >
   <name>vmx-11</name>
   <maxCoresPerSocket>128</maxCoresPerSocket>
   <supportedOperatingSystems>
      <OperatingSystemFamilyInfo>
         <Name>Microsoft Windows</Name>
         <OperatingSystemFamilyId>1</OperatingSystemFamilyId>
         <OperatingSystem>
            <DefaultHardDiskAdapterType ref="VirtualLsiLogicSASController">4
            </DefaultHardDiskAdapterType>
            <SupportedHardDiskAdapter ref="ParaVirtualSCSIController" />
            <SupportedHardDiskAdapter ref="VirtualLsiLogicController" />
            <SupportedHardDiskAdapter ref="VirtualLsiLogicSASController" />
            <SupportedHardDiskAdapter ref="VirtualIDEController" />
            <SupportedHardDiskAdapter ref="VirtualAHCIController" />
            <MinimumHardDiskSizeGigabytes>40
            </MinimumHardDiskSizeGigabytes>
            <MinimumMemoryMegabytes>512</MinimumMemoryMegabytes>
            <Name>Microsoft Windows Server 2016 (64-bit)</Name>
            <InternalName>windows9Server64Guest</InternalName>
            <Supported>true</Supported>
            <x64>true</x64>
            <MaximumCpuCount>128</MaximumCpuCount>
            <MaximumCoresPerSocket>128</MaximumCoresPerSocket>
            <MaximumSocketCount>128</MaximumSocketCount>
            <MinimumHardwareVersion>10</MinimumHardwareVersion>
            <PersonalizationEnabled>true</PersonalizationEnabled>
            <PersonalizationAuto>true</PersonalizationAuto>
            <SysprepPackagingSupported>true</SysprepPackagingSupported>
            <SupportsMemHotAdd>true</SupportsMemHotAdd>
            <cimOsId>102</cimOsId>
            <CimVersion>0</CimVersion>
            <SupportedForCreate>true</SupportedForCreate>
            <RecommendedNIC
               id="6"
               name="E1000E" />
            <SupportedNICType
               id="3"
               name="E1000" />
            <SupportedNICType
               id="6"
               name="E1000E" />
            <SupportedNICType
               id="3"
               name="E1000" />
            <SupportedNICType
               id="5"
               name="VMXNET3" />
         </OperatingSystem>
      </OperatingSystemFamilyInfo>
   </supportedOperatingSystems>
</VirtualHardwareVersion>
```

Update Multiple Sections of a Virtual Machine

You can make a single request that updates the `name`, `Description`, and any or all of the `VirtualHardwareSection`, `OperatingSystemSection`, `NetworkConnectionSection`, `GuestCustomizationSection` elements of a virtual machine.

Every `Vm` element contains a link to a `reconfigureVm` operation that you can use to update the `name`, `Description`, and multiple sections in a single operation. Sections that you omit from the request body are not updated.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the `Vm` element that you want to update.

2 Modify the retrieved Vm element.

Modifications can include the name, Description, and any or all of the VirtualHardwareSection, OperatingSystemSection, NetworkConnectionSection, GuestCustomizationSection elements of the Vm.

Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. Link elements and href attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3 Use the modified Vm as the body of a reconfigureVm request.

Results

The modified Vm replaces the contents of the retrieved Vm. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Update Multiple Sections of a Virtual Machine

This example uses the reconfigureVm operation to accomplish the updates shown in Update a NetworkConnectionSection and Modify the Guest Customization Section of a Virtual Machine in a single operation. It also updates the independent disk attached to this virtual machine in a way similar to the operation shown in Update an Independent Disk

Request:

```
POST https://vcloud.example.com/api/vApp/vm-4/action/reconfigureVm
Content-Type: application/vnd.vmware.vcloud.vm+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Vm ...>
   ...
   <NetworkConnectionSection
      type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
      xmlns="http://www.vmware.com/vcloud/v1.5"
      xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
      <ovf:Info>Firewall allows access to this address.</ovf:Info>
      <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
      <NetworkConnection network="vAppNetwork">
         <NetworkConnectionIndex>0</NetworkConnectionIndex>
         <IpAddress>10.147.115.1</IpAddress>
         <IsConnected>true</IsConnected>
         <MACAddress>00:50:56:01:01:49</MACAddress>
         <IpAddressAllocationMode>STATIC</IpAddressAllocationMode>
      </NetworkConnection>
   </NetworkConnectionSection>
   <GuestCustomizationSection
      xmlns="http://www.vmware.com/vcloud/v1.5"
      xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
      ovf:required="false">
      <ovf:Info>Specifies Guest OS Customization Settings</ovf:Info>
      <Enabled>true</Enabled>
      <ChangeSid>true</ChangeSid>
      <VirtualMachineId>12</VirtualMachineId>
      <JoinDomainEnabled>false</JoinDomainEnabled>
      <UseOrgSettings>false</UseOrgSettings>
      <DomainName />
      <DomainUserName />
      <DomainUserPassword />
      <AdminPasswordEnabled>true</AdminPasswordEnabled>
      <AdminPasswordAuto>true</AdminPasswordAuto>
      <AdminPassword />
      <ResetPasswordRequired>false</ResetPasswordRequired>
      <CustomizationScript />
      <ComputerName>Win2K3</ComputerName>
   </GuestCustomizationSection>
   <ovf:VirtualHardwareSection
      xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
      vcloud:type="application/vnd.vmware.vcloud.virtualHardwareSection+xml">
      <ovf:Info>Virtual hardware requirements</ovf:Info>
      <ovf:Item>
         <rasd:AddressOnParent>0</rasd:AddressOnParent>
         <rasd:HostResource
            vcloud:storageProfileHref="https://vcloud.example.com/api/vdcStorageProfile/3"
            vcloud:disk="https://vcloud.example.com/api/disk/128" />
         <rasd:InstanceID>2000</rasd:InstanceID>
         <rasd:Parent>2</rasd:Parent>
         <rasd:ResourceType>17</rasd:ResourceType>
      </ovf:Item>
   </ovf:VirtualHardwareSection>
   ...
</Vm>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
   ...
   operation="Updating Virtual Application Win2K3 (4)"
   ...>
   ...
</Task>
```

Retrieve or Update a Modifiable Section

You can make a GET request to the URL of any modifiable section to retrieve it for modification. After you modify the section, you can make a PUT request to its `edit` link to update the section with your modifications.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the section to modify.

   Make a GET request to the URL in the section's `href` attribute value.

2 Modify the retrieved section.

   Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3 Update the section with your modifications.

   Find the `Link` element in the section where `rel="edit"`. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

   For most section types, the response to this request is a `Task` element that tracks the update operation. When the task completes, the section is updated.

Results

The modified section replaces the contents of the original section. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Retrieve a NetworkConfigSection

This example retrieves the `NetworkConfigSection` of the vApp shown in Configuration Links in a vApp.

Request:

```
GET https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection
```

Response:

```
200 OK
Content-type: application/vnd.vmware.vcloud.networkConfigSection+xml
...
<NetworkConfigSection
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/"
   ovf:required="false">
   <ovf:Info>Configuration parameters for logical networks</ovf:Info>
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.networkConfigSection+xml"
      href="https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/" />
   <NetworkConfig
      networkName="vAppNetwork">
      <Configuration>
         <IpScopes>
            <IpScope>
               <IsInherited>true</IsInherited>
               <Gateway>10.147.56.253</Gateway>
               <Netmask>255.255.255.0</Netmask>
               <Dns1>10.147.115.1</Dns1>
               <Dns2>10.147.115.2</Dns2>
               <DnsSuffix>example.com</DnsSuffix>
               <IpRanges>
                  <IpRange>
                     <StartAddress>10.147.56.1</StartAddress>
                     <EndAddress>10.147.56.255</EndAddress>
                  </IpRange>
               </IpRanges>
            </IpScope>
         </IpScopes>
         <ParentNetwork
            type="application/vnd.vmware.vcloud.network+xml"
            name="Internet"
            href="https://vcloud.example.com/api/network/54" />
         <FenceMode>bridged</FenceMode>
      </Configuration>
      <IsDeployed>false</IsDeployed>
   </NetworkConfig>
</NetworkConfigSection>
```

For an example that updates this section, see Update a NetworkConfigSection .

Update a vApp Network Configuration

To change the configuration of a vApp network, you retrieve the `NetworkConfigSection` element of the vApp, modify it, and use it with a PUT request to update the section.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the vApp's `NetworkConfigSection`.

2 Modify the returned NetworkConfigSection as needed.

   Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. Link elements and href attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3 Update the NetworkConfigSection in the vApp.

   Find the Link element in the section where rel="edit". Make a PUT request to the URL in that link's href attribute value, and supply the modified section as the request body.

Example: Update a NetworkConfigSection

This example modifies the NetworkConfigSection that was retrieved in Retrieve a NetworkConfigSection. The modifications change the FenceMode value to natRouted and add a Features element that defines several network features that are useful to an FTP server that must be reachable from the public Internet, but only at the FTP and SSH ports. The modifications add the following items:

- A set of FirewallRules that allow TCP traffic to ports 21 and 22. Because these rules require you to specify a single IP address on the inside of the firewall, the IpScope element is modified to limit the range of IP addresses available on the vApp network to a single address. Any virtual machine that connects to the vApp network defined in this NetworkConfigSection is given this address.
- A NatService element that maps a routable external IP address to the internal IP address allocated to the Vm by the vApp network. The VAppScopedVmId value in this element is taken from the VAppScopedLocalId element of the Vm and the VmNicId value is taken from its PrimaryNetworkConnectionIndex. See Configuration Links in a Vm Element.

For more information about these and other network services in vApp networks, see Network Services in vApp Networks

This request, like all request bodies derived from a response, omits the Link elements and href attributes that were part of the retrieved NetworkConfigurationSection. It also omits the IsDeployed element of the NetworkConfig. These elements and attributes are created by the server and are read-only. They are ignored if you include them in a request. Read-only elements are noted in the schema reference.

Request:

```
PUT https://vcloud.example.com/api/vApp/vapp-7/networkConfigSection/
Content-type: application/vnd.vmware.vcloud.networkConfigSection+xml
...
<NetworkConfigSection
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <ovf:Info>Configuration parameters for logical networks</ovf:Info>
   <NetworkConfig
      networkName="vAppNetwork">
      <Configuration>
         <IpScopes>
            <IpScope>
               <IsInherited>false</IsInherited>
               <Gateway>10.147.56.253</Gateway>
               <Netmask>255.255.255.0</Netmask>
               <Dns1>10.147.115.1</Dns1>
               <Dns2>10.147.115.2</Dns2>
               <DnsSuffix>example.com</DnsSuffix>
               <IpRanges>
                  <IpRange>
                     <StartAddress>10.147.56.1</StartAddress>
                     <EndAddress>10.147.56.1</EndAddress>
                  </IpRange>
               </IpRanges>
            </IpScope>
         </IpScopes>
         <ParentNetwork
            type="application/vnd.vmware.vcloud.network+xml"
            name="Internet"
            href="https://vcloud.example.com/api/network/54" />
         <FenceMode>natRouted</FenceMode>
         <Features>
            <FirewallService>
               <IsEnabled>true</IsEnabled>
               <FirewallRule>
                  <IsEnabled>true</IsEnabled>
                  <Description>FTP Rule</Description>
                  <Policy>allow</Policy>
                  <Protocols>
                     <Tcp>true</Tcp>
                  </Protocols>
                  <DestinationPortRange>21</DestinationPortRange>
                  <DestinationIp>10.147.115.1</DestinationIp>
                  <SourcePortRange>any</SourcePortRange>
                  <SourceIp>any</SourceIp>
                  <EnableLogging>false</EnableLogging>
               </FirewallRule>
               <FirewallRule>
                  <IsEnabled>true</IsEnabled>
                  <Description>SSH Rule</Description>
                  <Policy>allow</Policy>
                  <Protocols>
                     <Tcp>true</Tcp>
                  </Protocols>
                  <DestinationPortRange>22</DestinationPortRange>
                  <DestinationIp>10.147.115.1</DestinationIp>
                  <SourcePortRange>any</SourcePortRange>
                  <SourceIp>any</SourceIp>
                  <EnableLogging>false</EnableLogging>
               </FirewallRule>
            </FirewallService>
            <NatService>
               <IsEnabled>true</IsEnabled>
               <NatType>ipTranslation</NatType>
               <Policy>allowTraffic</Policy>
               <NatRule>
                  <OneToOneVmRule>
                     <MappingMode>automatic</MappingMode>
                     <VAppScopedVmId>3963994b-5a0a-48fe-b9ae-7f9a2d8e8e5b</VAppScopedVmId>
                     <VmNicId>0</VmNicId>
                  </OneToOneVmRule>
               </NatRule>
            </NatService>
         </Features>
      </Configuration>
   </NetworkConfig>
</NetworkConfigSection>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
   ...
</Task>
```

Important Whenever you modify a vApp network, as we do in this example, you must be sure that the modifications are consistent with the network connection requirements of the virtual machines in the vApp. The vApp in this example contains a single virtual machine. Its `NetworkConnection` element, shown in Configuration Links in a Vm Element, specifies an IP address that will not be available after the vApp network is reconfigured as shown here. Update a NetworkConnectionSection corrects this problem. This example uses the `IpScope` element to restrict the IP addresses available on a vApp network. It is usually more practical to use a wide range of addresses available on a vApp network and apply any firewall-related IP address restrictions by modifying the `NetworkConnectionSection` of the `Vm` to which the `FirewallRules` apply, as shown in Update a NetworkConnectionSection. A wider range of IP addresses allows you to modify this vApp to include additional virtual machines, and the IP address restriction applied in Update a NetworkConnectionSection allows the `FirewallRules` in this example to remain valid.

Update the NetworkConnectionSection of a Virtual Machine

Whenever you create a vApp network or update its configuration, you might also need to update the `NetworkConnectionSection` elements of the virtual machines in the vApp.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the virtual machine's `NetworkConnectionSection`.

2 Modify the returned `NetworkConnectionSection` as needed.

   Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3 Update the section with your modifications.

   a In the retrieved section, find the `Link` element where `rel="edit"`.

   b Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

   The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

Example: Update a NetworkConnectionSection

This example modifies the `NetworkConnectionSection` shown in Configuration Links in a Vm Element so that this network connection is compatible with the reconfigured vApp network to which it must connect. See Update a NetworkConfigSection. The modified `NetworkConnectionSection` in the request changes two values:

- The `IpAddress` now specifies the address to which the vApp network's firewall allows access.
- Because it specifies an IP address, the modified `NetworkConnectionSection` also changes the value of the `IpAddressAllocationMode` from `DHCP` to `MANUAL`.

Note The `ovf:Info` element is a required member of `NetworkConnectionSection` and all other sections that are derived from `ovf:SectionType`. The element must be present, even if it has no content. In this example, we use the content to explain why the connection is configured this way.

Request:

```
PUT "https://vcloud.example.com/api/vApp/vm-4/networkConnectionSection/
Content-type: application/vnd.vmware.vcloud.networkConnectionSection+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<NetworkConnectionSection
   type="application/vnd.vmware.vcloud.networkConnectionSection+xml"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <ovf:Info>Firewall allows access to this address.</ovf:Info>
   <PrimaryNetworkConnectionIndex>0</PrimaryNetworkConnectionIndex>
   <NetworkConnection
```

```
      network="vAppNetwork">
      <NetworkConnectionIndex>0</NetworkConnectionIndex>
      <IpAddress>10.147.115.1</IpAddress>
      <IsConnected>true</IsConnected>
      <MACAddress>00:50:56:01:01:49</MACAddress>
      <IpAddressAllocationMode>MANUAL</IpAddressAllocationMode>
   </NetworkConnection>
</NetworkConnectionSection>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
   ...
</Task>
```

Retrieve or Modify the CPU Configuration of a Virtual Machine

The CPU configuration of a virtual machine is represented by an `Item` in its `VirtualHardwareSection` element.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the CPU section to modify.

Make a GET request to the URL in the section's `href` attribute value:

```
GET https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu
```

2 Modify the retrieved section.

Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3 Update the section with your modifications.

a In the retrieved section, find the `Link` element where `rel="edit"`.

b Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

Results

The modified section replaces the contents of the original section. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Modify the CPU Configuration of a Virtual Machine

The initial configuration for the virtual machine used in this example shows a single CPU and `CoresPerSocket` value of 1.

```
<Item xmlns="http://schemas.dmtf.org/ovf/envelope/1"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
   xmlns:vmw="http://www.vmware.com/schema/ovf"
   vcloud:type="application/vnd.vmware.vcloud.rasdItem+xml">
   <rasd:AllocationUnits>hertz * 10^6</rasd:AllocationUnits>
   <rasd:Description>Number of Virtual CPUs</rasd:Description>
   <rasd:ElementName>1 virtual CPU(s)</rasd:ElementName>
   <rasd:InstanceID>4</rasd:InstanceID>
   <rasd:Reservation>0</rasd:Reservation>
   <rasd:ResourceType>3</rasd:ResourceType>
   <rasd:VirtualQuantity>1</rasd:VirtualQuantity>
   <rasd:Weight>0</rasd:Weight>
   <vmw:CoresPerSocket ovf:required="false">1</vmw:CoresPerSocket>
</Item>
```

This request modifies the CPU section to add a second CPU to the `Vm` by changing the `rasd:VirtualQuantity` value of the `Item` to `2`. It also raises the value of `CoresPerSocket` to 2.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/cpu
Content-type: application/vnd.vmware.vcloud.rasdItem+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Item xmlns="http://schemas.dmtf.org/ovf/envelope/1"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
   vcloud:type="application/vnd.vmware.vcloud.rasdItem+xml">
   <rasd:AllocationUnits>hertz * 10^6</rasd:AllocationUnits>
   <rasd:Description>Number of Virtual CPUs</rasd:Description>
   <rasd:ElementName>2 virtual CPU(s)</rasd:ElementName>
   <rasd:InstanceID>4</rasd:InstanceID>
   <rasd:Reservation>0</rasd:Reservation>
   <rasd:ResourceType>3</rasd:ResourceType>
   <rasd:VirtualQuantity>2</rasd:VirtualQuantity>
   <rasd:Weight>0</rasd:Weight>
   <vmw:CoresPerSocket ovf:required="false">2</vmw:CoresPerSocket>
</Item>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>
```

Retrieve or Modify the GuestCustomizationSection of a Virtual Machine

The `GuestCustomizationSection` element includes a customization script and other parameters that are applied when you customize a virtual machine.

The `GuestCustomizationSection` includes predefined property names that VMware guest customization tools recognize. Certain values in this element, if omitted or left empty, are inherited from the `OrgGuestPersonalizationSettings` of the organization that owns the virtual machine. See Retrieve or Update Organization Settings.

The VMware Cloud Director API also supports use of the `ovf:ProductSection` to pass an arbitrary set of *key*=*value* pairs to a vApp or virtual machine through the `ovf:Environment` element. See Retrieve or Modify ProductSection Elements.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the `GuestCustomizationSection` to modify.

   Make a GET request to the URL in the section's `href` attribute value.

   ```
   GET https://vcloud.example.com/api/vApp/vm-12/guestCustomizationSection/
   ```

2 Modify the retrieved section.

   Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3 Update the section with your modifications.

   a In the retrieved section, find the `Link` element where `rel="edit"`.

   b Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

   The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

Results

The modified section replaces the contents of the original section. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Modify the Guest Customization Section of a Virtual Machine

This request specifies guest customization values, including the information required to join the virtual machine to a Windows domain.

Note If you include a `CustomizationScript`, it cannot exceed 49,000 characters.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-12/guestCustomizationSection/
Content-type: application/vnd.vmware.vcloud.guestcustomizationsection+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<GuestCustomizationSection
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   ovf:required="false">
   <ovf:Info>Specifies Guest OS Customization Settings</ovf:Info>
   <Enabled>true</Enabled>
   <ChangeSid>true</ChangeSid>
   <VirtualMachineId>12</VirtualMachineId>
   <JoinDomainEnabled>false</JoinDomainEnabled>
   <UseOrgSettings>false</UseOrgSettings>
   <DomainName>example</DomainName>
   <DomainUserName>admin</DomainUserName>
   <DomainUserPassword>Pa55w0rd</DomainUserPassword>
   <AdminPasswordEnabled>true</AdminPasswordEnabled>
   <AdminPasswordAuto>true</AdminPasswordAuto>
   <AdminPassword />
   <ResetPasswordRequired>false</ResetPasswordRequired>
   <CustomizationScript />
   <ComputerName>Win2K3</ComputerName>
</GuestCustomizationSection>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
   ...
   operation="Updating Virtual Application Win2K3 (12)"
   ...>
   ...
</Task>
```

Retrieve or Modify ProductSection Elements

`ProductSection` elements allow you to pass runtime information to vApps and virtual machines. The *`key=value`* pairs in this section are made available in the OVF `Environment` of a powered-on vApp or virtual machine.

A vApp or virtual machine can get runtime information from its `ovf:Environment` element. This read-only element is populated with information from a `ProductSection` element when the vApp or virtual machine is powered on. A `Vm` can use VMware Tools to read these values from its `ovf:Environment`. A `Vm` can also read the values by mounting a special media object. To make a *`key=value`* pair available in the `ovf:Environment`, add it to the appropriate `ProductSection` of a vApp template or powered-off vApp or virtual machine.

Note All `ProductSection` elements in a vApp template, vApp, or virtual machine are returned as members of a `ProductSectionList`. You cannot retrieve or update an individual `ProductSection`. You must retrieve and modify the `ProductSectionList` to update the individual `ProductSection` elements it contains.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the `ProductSectionList` from the vApp or virtual machine.

   Use a request like this one, which targets a vApp.

   ```
   GET https://vcloud.example.com/api/vApp/vapp-123/productSections
   ```

   The response is a `ProductSectionList` element, which contains all the `ProductSection` elements in the vApp, along with a `Link` element that contains the `rel="edit"` URL to use when updating the `ProductSectionList`. If the vApp contains no `ProductSection` elements, the response contains only the `Link` element.

2 Modify the retrieved `ProductSectionList`.

   You can modify existing `ProductSection` elements, create new ones, or both. `ProductSection` has no required contents. Unlike updates to other sections, updates to a `ProductSection` merge new and existing values, subject to the following rules:

   - `Property` elements that are present in the existing `ProductSection` but not in the update are removed.
   - `Property` elements that are present in the update but not in the in the existing `ProductSection` are added to the `ProductSection` if they have a corresponding `Value` element.
   - If a `Property` element that is present in the existing `ProductSection` has different attributes, qualifiers, or other details in the update, the `Property` in the update replaces the existing one.
   - If a `Property` element that is present in the existing `ProductSection` has no `Value` in the update, the existing `Property` and `Value` remain unchanged.

3 Update the section with your modifications.

   Find the `Link` element in the `ProductSectionList` where `rel="edit"`. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified `ProductSectionList` as the request body.

   The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

Results

The modified section replaces the contents of the original section, subject to the rules listed in Step 2.

Example: Update a ProductSection in a vApp

This request creates or updates a `ProductSectionList` that contains a single `ProductSection`. The `ProductSection` sets three properties. The response is a `Task`.

Request:

```
PUT https://vcloud.example.com/api/vApp/vapp-123/productSections
Content-Type: application/vnd.vmware.vcloud.productSections+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ProductSectionList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <ovf:ProductSection
      required="true">
      <ovf:Info>Information about the installed software</ovf:Info>
      <ovf:Property
         ovf:type="string"
         ovf:key="CRM_Database_Host"
         ovf:value="CRM.example.com">
         <ovf:Label>CRM Database Host</ovf:Label>
      </ovf:Property>
      <ovf:Property
         ovf:type="string"
         ovf:key="CRM_Database_Username"
         ovf:value="dbuser">
         <ovf:Label>CRM Database Usernname</ovf:Label>
      </ovf:Property>
      <ovf:Property
         ovf:type="string"
         ovf:key="CRM_Password"
         ovf:value="Pa55w0rd">
         <ovf:Label>CRM Database User Password</ovf:Label>
      </ovf:Property>
   </ovf:ProductSection>
</ProductSectionList>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application ..." ...>
...
</Task>
```

After the vApp is powered on, a virtual machine can retrieve the `ovf:Environment` document in the following ways:

- It can use the default OVF `iso` transport type. This makes the environment document available as a file named `ovf-env.xml` on an ISO image that is mounted on the first available CD-ROM device on the virtual machine. You can use any convenient mechanism to read this file.

  ```
  [root@example-vm-RHEL ~] cat /media/cdrom/ovf-env.xml
  <?xml version="1.0" encoding="UTF-8"?>
  <Environment ...>
     ...
     <PropertySection>
        <Property
           oe:key="CRM_Database_Host"
           oe:value="CRM.example.com" />
        <Property
           oe:key="CRM_Database_Username"
           oe:value="dbuser" />
        <Property
           oe:key="CRM_Password"
           oe:value="Pa55w0rd" />
      </PropertySection>
     ...
  </Environment>
  ```

- If the virtual machine has VMware Tools installed, it can use the vmtoolsd program, as shown here.

  ```
  [root@example-vm-RHEL ~] /usr/sbin/vmtoolsd --cmd 'info-get guestinfo.ovfEnv'
  <?xml version="1.0" encoding="UTF-8"?>
  <Environment ...>
     ...
     <PropertySection>
        <Property
           oe:key="CRM_Database_Host"
           oe:value="CRM.example.com" />
        <Property
           oe:key="CRM_Database_Username"
           oe:value="dbuser" />
        <Property
           oe:key="CRM_Password"
           oe:value="Pa55w0rd" />
      </PropertySection>
     ...
  </Environment>
  ```

  On Windows, the vmtoolsd executable file is typically installed in C:\Program Files\VMware\VMware Tools\vmtoolsd.exe

Retrieve or Modify Groups of Related Sections in a Virtual Machine

The VMware Cloud Director API provides links that you can use to retrieve or update groups of sections that define related hardware items such as disks, media devices, and network cards in a Vm element.

As shown in Configuration Links in a Vm Element, Link elements for disks, media devices, and network cards are grouped at the end of the VirtualHardwareSection. These links have content type application/vnd.vmware.vcloud.rasdItemsList+xml, and reference a RasdItemsList element in the VirtualHardwareSection of a Vm. The VMware Cloud Director API uses the RasdItemsList element to aggregate related elements in a VirtualHardwareSection. This approach simplifies retrieval and modification of Item elements that are typically viewed or modified as a group.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the `RasdItemsList` from a `Vm`.

  Make a GET request to the URL in the link where `type="application/vnd.vmware.vcloud.rasdItemsList+xml"` and `rel="down"`. See Retrieve the Hard Disks and Controllers in a Virtual Machine .

2 Modify the items in the retrieved list.

  Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3 Update the sections with your modifications.

  Make a PUT request to the URL in the link where `type="application/vnd.vmware.vcloud.rasdItemsList+xml"` and `rel="edit"`, and supply the modified section as the request body.

  The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

Example: Retrieve the Hard Disks and Controllers in a Virtual Machine

This example uses the `virtualHardwareSection/disks` link shown in Configuration Links in a Vm Element to retrieve the list of hard disks and hard disk controllers for a virtual machine.

Request:

```
GET https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.rasdItemsList+xml
...
<RasdItemsList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
   type="application/vnd.vmware.vcloud.rasdItemsList+xml"
   href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks" ... >
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.rasdItemsList+xml"
      href="https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks" />
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>SCSI Controller</rasd:Description>
      <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
      <rasd:InstanceID>2</rasd:InstanceID>
      <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
      <rasd:ResourceType>6</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>0</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 1</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="1024"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2000</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>1</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 2</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="2048"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2001</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>IDE Controller</rasd:Description>
      <rasd:ElementName>IDE Controller 0</rasd:ElementName>
      <rasd:InstanceID>3</rasd:InstanceID>
      <rasd:ResourceType>5</rasd:ResourceType>
   </Item>
</RasdItemsList>
```

Retrieve or Modify the Hard Disk Configuration of a Virtual Machine

The hard disk configuration of a virtual machine is represented by one or more `Item` elements in its `VirtualHardwareSection`.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the hard disk configuration from the virtual machine.

Make a GET request to the virtual machine's `virtualHardwareSection/disks` link.

```
GET https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
```

The response to this kind of request is a `RasdItemsList` element that contains an `Item` element for each of the virtual machine's hard disks and hard disk controllers, as shown in Retrieve the Hard Disks and Controllers in a Virtual Machine .

Important If an independent disk is attached to the virtual machine, it is included in this list, but cannot be modified by this operation. Attached independent disks are distinguished by the appearance of a `vcloud:disk` attribute in the containing `Item`, as shown here:

```
<rasd:HostResource
   ...
   vcloud:disk="https://vcloud.example.com/api/disk/128" />
```

If you need to modify an independent disk while it is attached to a virtual machine, you must use the `reconfigureVm` operation. See Update Multiple Sections of a Virtual Machine.

2 Modify the retrieved section.

Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

You cannot modify the values of the `busType` and `busSubType` attributes after you create a new disk. When creating a new disk, be sure to set the values of `busType` and `busSubType` to a valid combination.

Table 6-2. Valid Combinations of busType and busSubType

busType	busSubType	Controller
5	null	IDE controller
6	`buslogic`	BusLogic Parallel SCSI controller
6	`lsilogic`	LSI Logic Parallel SCSI controller
6	`lsilogicsas`	LSI Logic SAS SCSI controller

Table 6-2. Valid Combinations of busType and busSubType (continued)

busType	busSubType	Controller
6	`VirtualSCSI`	Paravirtual SCSI controller
20	`vmware.sata.ahci`	SATA controller (hardware version 10 and later)

Note If you remove all the hard disk objects (RASD resource type 17) from the `RasdItemsList` container for disks in the `VirtualHardwareSection`, the system also removes all hard disk controllers (RASD resource type 5) from that section.

3 Update the section with your modifications.

   a In the retrieved section, find the `Link` element where `rel="edit"`.

   b Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

   The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

Results

The modified section replaces the contents of the original section. For some section types, modifications take effect immediately. For others, modifications take effect only after a power or deployment state change.

Example: Modify the Hard Disk Configuration of a Virtual Machine

The following request increases the capacity of the hard disk from 1GB to 10GB by changing the `vcloud:capacity` value of the `Item` that defines the disk. The capacity is raised from `1024` to `10240`. The request body includes the entire `RasdItemsList` returned by the request shown in Step 1, even though only one element is changed. `Link` elements from a response are ignored if you include them in a request, so they are omitted in this example.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
Content-Type: application/vnd.vmware.vcloud.rasditemslist+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<RasdItemsList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
   type="application/vnd.vmware.vcloud.rasdItemsList+xml" >
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>SCSI Controller</rasd:Description>
      <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
      <rasd:InstanceID>2</rasd:InstanceID>
      <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
      <rasd:ResourceType>6</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>0</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 1</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="10240"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2000</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>1</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 2</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="2048"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2001</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>IDE Controller</rasd:Description>
      <rasd:ElementName>IDE Controller 0</rasd:ElementName>
      <rasd:InstanceID>3</rasd:InstanceID>
      <rasd:ResourceType>5</rasd:ResourceType>
   </Item>
</RasdItemsList>
```

The response is a task.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>
```

Retrieve or Update the VDC Compute Policy of a Virtual Machine

You can change the VDC compute policy of a virtual machine by sending a PUT request to the `reconfigureVm` link of the virtual machine.

For information about VDC compute policies, see Managing VM-Host Affinity Rules.

Prerequisites

- You must be a **system administrator**, **organization administrator**, **vApp Author**, or **vApp User**.
- The **system administrator** published the new VDC compute policy to the organization VDC of the target virtual machine.

Procedure

1 Retrieve the `Vm` element of the target virtual machine.

For example, use a request similar to:

```
GET https://vcloud.example.com/api/vApp/vm-75
```

The `VdcComputePolicy` element references the current VDC compute policy of the virtual machine.

2 To update the VDC compute policy, modify the retrieved `Vm` element by editing the `VdcComputePolicy` reference.

For example, modify the `Vm` element similar to:

```
<Vm
   xmlns="http://www.vmware.com/vcloud/v1.5"
   ...
      <VCloudExtension required="false">
   ...
   <VdcComputePolicy
      href="https://vcloud.example.com/cloudapi/1.0.0/vdcComputePolicies/51"
      id="51"
      name="oracle_license"
      type="application/json"/>
   ...
</Vm>
```

3 Use the modified `Vm` element as the body of a `reconfigureVm` request.

For example, use a request similar to:

```
PUT https://vcloud.example.com/api/vApp/vm-75/action/reconfigureVm
type="application/vnd.vmware.vcloud.vm+xml
...
<Vm
   ...
   <VdcComputePolicy
      .../>
   ...
</Vm>
```

Results

The placement engine moves the virtual machine to the corresponding VM group of the new VDC compute policy.

Update the Storage Profile for a Virtual Machine

You can update a `Vm` to revalidate the storage profile it uses or specify a different storage profile. Revalidation of a virtual machine's current storage profile is required whenever the datastore that supports the virtual machine changes.

Every `Vm` element includes a `StorageProfile` element whose `href` attribute value specifies the default storage profile consumed by the virtual machine. This default is used for all hard disk`Item`s in the `VIrtualHardwareSection` that do not specify `storageProfileOverrideVmDefault`.

If you do not specify a `StorageProfile` during an instantiate, compose, or recompose operation, it is inherited from the organization VDC in which the virtual machine is deployed. To change the value of an existing `StorageProfile`, you must update the entire `Vm` element that contains it.

Important When the system administrator changes the datastore that stores a virtual machine, you must update the `Vm` element as shown in Update the Storage Profile for a Virtual Machine.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1. Retrieve the `Vm` element.

   Make a GET request to the URL in the value of the `href` attribute of the `Vm`.

2. Modify the retrieved `Vm` to change the `StorageProfile` reference.

   Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3. Update the `Vm` with your modifications.

   a. Find the `Link` element in the `Vm` where `rel="edit"`.

   b. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified `Vm` as the request body.

The response to this request is a Task element that tracks the relocation of the virtual machine to a datastore in the new storage profile. When the task is complete, the virtual machine's StorageProfile has been updated and the virtual machine has been relocated to the new storage profile.

Example: Update the Storage Profile for a Virtual Machine

This example shows a Vm element containing a StorageProfile. The actual update operation requires the entire Vm element, including the StorageProfile, in the request body. Only a small part of the element appears in this example.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4
Content-type: application/vnd.vmware.vcloud.vm+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Vm ...>
...
 <StorageProfile
      type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
      name="Gold"
      href="https://vcloud.example.com/api/vdcStorageProfile/3" />

</Vm>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>
```

Override the Default Storage Profile for a Hard Disk

By default, all hard disks defined in the VirtualHardwareSection of a Vm element use the storage profile specified for the Vm. You can override this default for any of these disks when you instantiate a vApp template, compose or recompose a vApp, or reconfigure a virtual machine.

Every `Vm` element includes a `StorageProfile` element. The storage profile referenced in this element normally provides storage for all the hard disk `Item`s in the virtual machine's `VirtualHardwareSection`. You can override this default by updating the virtual machine's `VirtualHardwareSection` to add `storageProfileOverrideVmDefault` and `storageProfileHref` attributes to the `Item` that defines the hard disk. You can update a `VirtualHardwareSection` when you are instantiating a vApp template, composing or recomposing a vApp, or reconfiguring a virtual machine.

Important You cannot override the default storage profile for any hard disk of a virtual machine that is deployed in a VDC where fast provisioning is enabled.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Procedure

1 Retrieve the hard disk configuration from the virtual machine.

Make a GET request to the virtual machine's `virtualHardwareSection/disks` link.

```
GET https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
```

The response to this kind of request is a `RasdItemsList` element that contains an `Item` element for each of the virtual machine's hard disks and hard disk controllers, as shown in Retrieve the Hard Disks and Controllers in a Virtual Machine .

2 In the `VirtualHardwareSection` of the retrieved `Vm`, modify the `rasd:HostResource` element of the `Item` that defines the disk for which you want to override the default storage profile.

Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

a In the `VirtualHardwareSection` of the `Vm`, find the `Item` that represents the hard disk for which you want to override the default storage profile.

b Add a `storageProfileHref` to the `rasd:HostResource` element of the `Item` and set its value to the `href` of the storage profile you want to use for this disk. The storage profile you specify must be available in the VDC where this virtual machine is deployed.

c Add a `storageProfileOverrideVmDefault` attribute to the `rasd:HostResource` element of the `Item`. The value of this attribute controls whether changes to the virtual machine's `StorageProfile` affect the storage profile that this disk uses.

Table 6-3. How `storageProfileOverrideVmDefault` Values Affect Hard Disk Storage Profile Assignment

Value	Result
`true` (default)	The storage profile specified in `storageProfileHref` is always used for this disk regardless of the value specified in the virtual machine's `StorageProfile`
`false`	Any storage profile that is specified in `storageProfileHref` is ignored and the disk is migrated to the storage specified in the virtual machine's `StorageProfile`.

If you omit the `storageProfileOverrideVmDefault` attribute, the `storageProfileHref` is ignored.

3 Update the section with your modifications.

a In the retrieved section, find the `Link` element where `rel="edit"`.

b Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

Example: Override the Default Storage Profile for a Hard Disk

This example builds on the ones shown in Update the Storage Profile for a Virtual Machine and Modify the Hard Disk Configuration of a Virtual Machine. This virtual machine has the storage profile specified in Update the Storage Profile for a Virtual Machine:

```
<StorageProfile
   type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
   name="Gold"
   href="https://vcloud.example.com/api/vdcStorageProfile/3" />
```

To specify a new storage profile for the disk whose capacity was increased in Modify the Hard Disk Configuration of a Virtual Machine, you must provide the `storageProfileHref` for the new storage profile and also set the `storageProfileOverrideVmDefault` attribute to true.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
Content-Type: application/vnd.vmware.vcloud.rasditemslist+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<RasdItemsList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
   type="application/vnd.vmware.vcloud.rasdItemsList+xml" >
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>SCSI Controller</rasd:Description>
      <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
      <rasd:InstanceID>2</rasd:InstanceID>
      <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
      <rasd:ResourceType>6</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>0</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 1</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="10240"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"
         vcloud:storageProfileOverrideVmDefault="true"
         vcloud:storageProfileHref="https://vcloud.example.com/api/vdcStorageProfile/5">
      </rasd:HostResource>
      <rasd:InstanceID>2000</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>1</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 2</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="2048"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"></rasd:HostResource>
      <rasd:InstanceID>2001</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>IDE Controller</rasd:Description>
      <rasd:ElementName>IDE Controller 0</rasd:ElementName>
      <rasd:InstanceID>3</rasd:InstanceID>
      <rasd:ResourceType>5</rasd:ResourceType>
   </Item>
</RasdItemsList>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>
```

Specify Hard Disk IOPS

You can specify a desired level of read/write performance for a hard disk by including a `vcloud:iops` attribute in the OVF `Item` that represents the disk configuration.

Managed read/write performance in physical storage devices and virtual disks is defined in units called IOPS, which measure read/write operations per second. When an organization VDC storage profile is backed by a Provider VDC storage profile that includes storage devices that are capable of IOPS allocation, you can configure disks that use it to request a specified level of I/O performance. A storage profile configured with IOPS support delivers its default IOPS value to all disks that use it, even disks that are not configured to request a specific IOPS value. A hard disk configured to request a specific IOPS value cannot use a storage profile whose maximum IOPS value is lower than the requested value, or a storage profile that is not configured with IOPS support.

Note VMware Cloud Director sets an IOPS limit and reservation for every disk that uses an IOPS-enabled storage profile. vSphere is responsible for allocating the IOPS capacity of the underlying datastore across all virtual disks that use the storage profile. IOPS management is primarily intended to ensure that no disk can consume more than its fair share of IOPS. Realized IOPS for a given disk are limited by what the backing LUN can provide, and can be influenced by factors such as read/write block size. While a given storage profile can include a mix of datastores that IOPS-enabled and those that are not, such configurations can interfere with the system's ability to allocate IOPS fairly across all disks that use the storage profile.

Specify hard disk IOPS only when you have a well-defined need for a specific level of disk performance, and are confident that all storage profiles that the disk is likely to use can provision the desired level of IOPS. Because requesting a specific IOPS value imposes limitation on the set of storage profiles that a virtual machine can use, it is a best practice to avoid specifying hard disk IOPS in cases where the disk or virtual machine is likely to migrate to environments where an appropriate storage profile is not available.

Prerequisites

This operation requires the rights included in the predefined vApp Author role or an equivalent set of rights.

Verify that the disk can use a storage profile configured to support IOPS. See Configure Storage I/O Control Support in an Organization VDC.

Procedure

1 Retrieve the hard disk configuration from the virtual machine.

   Make a GET request to the virtual machine's `virtualHardwareSection/disks` link.

   ```
   GET https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
   ```

   The response to this kind of request is a `RasdItemsList` element that contains an `Item` element for each of the virtual machine's hard disks and hard disk controllers.

2 Verify that the hard disk uses a storage profile that is configured with IOPS support.

By default, all hard disks defined in the `VirtualHardwareSection` element of a virtual machine use the storage profile specified for the virtual machine. Some or all of the disks in a virtual machine can override this default and specify their own storage profile.

a Determine whether any of the disks override the virtual machine default storage profile.

All disks that do not use the default storage profile include `vcloud:storageProfileOverrideVmDefault` and `vcloud:storageProfileHref` attributes, as shown in this example.

```
<Item>
   ...
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="10240"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"
         vcloud:storageProfileOverrideVmDefault="true"
         vcloud:storageProfileHref="https://vcloud.example.com/api/
vdcStorageProfile/5">
   ...
</Item>
```

Retrieve the storage profile by making a GET request to the URL in the `vcloud:storageProfileHref` attribute.

b Disks that do not override the default storage profile use the one defined by the virtual machine.

Retrieve the `Vm` and examine the response to find its `StorageProfile` element. Retrieve the storage profile by making a GET request to the URL in the `href` attribute of the `StorageProfile` element.

A storage profile that is configured with IOPS support includes an `IopsSettings` element like the one shown here:

```
<VdcStorageProfile>
   ...
   <IopsSettings>
      <Enabled>true</Enabled>
      <DiskIopsMax>4000</DiskIopsMax>
      <DiskIopsDefault>1000</DiskIopsDefault>
      <StorageProfileIopsLimit>200000</StorageProfileIopsLimit>
      <DiskIopsPerGbMax>100</DiskIopsPerGbMax>
   </IopsSettings>
</VdcStorageProfile>
```

3 In the `VirtualHardwareSection` element that you retrieved in Step 1, modify the `rasd:HostResource` element of the `Item` that defines the disk for which you want to specify IOPS to add a `vcloud:iops` attribute.

   The value of the `vcloud:iops` attribute must be between 200 and 4000, and cannot be greater than the value of `DiskIopsMax` for the disk's storage profile. See Specify Hard Disk IOPS.

   Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

4 Update the section with your modifications.

   a In the retrieved section, find the `Link` element where `rel="edit"`.

   b Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body.

   The response to this request is a `Task` element that tracks the update operation. When the task is complete, the section is updated.

Example: Specify Hard Disk IOPS

This example is similar to the ones shown in Update the Storage Profile for a Virtual Machine and Modify the Hard Disk Configuration of a Virtual Machine but adds a `vcloud:iops` attribute to the `HostResource` that defines the disk. For the purpose of this example, assume that the virtual machine's default storage profile is enabled to provide IOPS support and does not place a lower limit on disk IOPS than the one requested.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4/virtualHardwareSection/disks
Content-Type: application/vnd.vmware.vcloud.rasditemslist+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<RasdItemsList
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
   type="application/vnd.vmware.vcloud.rasdItemsList+xml" >
   <Item>
      <rasd:Address>0</rasd:Address>
      <rasd:Description>SCSI Controller</rasd:Description>
      <rasd:ElementName>SCSI Controller 0</rasd:ElementName>
      <rasd:InstanceID>2</rasd:InstanceID>
      <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
      <rasd:ResourceType>6</rasd:ResourceType>
   </Item>
   <Item>
      <rasd:AddressOnParent>0</rasd:AddressOnParent>
      <rasd:Description>Hard disk</rasd:Description>
      <rasd:ElementName>Hard disk 1</rasd:ElementName>
      <rasd:HostResource
         xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
         vcloud:capacity="10240"
         vcloud:busSubType="lsilogic"
         vcloud:busType="6"
         vcloud:iops="1500"
      </rasd:HostResource>
      <rasd:InstanceID>2000</rasd:InstanceID>
      <rasd:Parent>2</rasd:Parent>
      <rasd:ResourceType>17</rasd:ResourceType>
   </Item>
   <Item>
      ...
   </Item>
   ...
</RasdItemsList>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>
```

Managing an Organization

7

The VMware Cloud Director API supports objects and operations that an organization administrator can use to automate tasks associated with managing organizations and the VDCs, networks, catalogs, and users they contain.

A successful login by an organization administrator returns a `Session` element, which contains a link that enables the administrator to retrieve a `VCloud` element. This element provides links to objects in the administrator's organization and read-only links to system-wide objects such as roles and rights. See Retrieve an Administrative View of a Cloud.

For a full list of the VMware Cloud Director API admin operations, see the *VMware Cloud Director API Schema Reference* at https://code.vmware.com.

Read the following topics next:

- Administrator Credentials and Privileges
- Organization Administration
- VDC Administration
- Network Administration
- Catalog Administration
- User and Group Administration
- About Federation and Single Sign-On
- Managing Rights and Roles

Administrator Credentials and Privileges

An administrator's privileges are scoped by the organization to which the administrator authenticates.

The VMware Cloud Director API defines two levels of administrative privilege:

- Organization administrators, who have administrative privileges in a specific organization.

- System administrators, who have superuser privileges throughout the system. System administrators are members of the `System` organization, and can create, read, update, and delete all objects in a cloud. They have organization administrator rights in all organizations in a cloud, and can operate directly on vSphere resources to create and modify provider VDCs, external networks, network pools, and similar system-level objects.

Some administrative operations, and all vSphere platform operations, are restricted to the system administrator. Before you attempt these operations, log in to the `System` organization with the user name and password of the system administrator account that was created when VMware Cloud Director was installed, or the user name and password of any member of the `System` organization. For example, a system administrator whose user name was defined as `administrator` would log in as `administrator@System`.

The System Organization

The `System` organization is created automatically when VMware Cloud Director is installed. Unlike the organizations represented by `Org` and `AdminOrg` objects, the `System` organization cannot contain catalogs, VDCs, groups, or users who are not system administrators.

The `System` organization is initially configured with one member, a local user defined as part of the VMware Cloud Director setup process. Like all organizations, the `System` organization is created with implicit support for the VMware Cloud Director integrated identity provider. A system administrator can reconfigure the `System` organization to use any of the other identity providers supported by VMware Cloud Director.

Example: The System Organization

When a system administrator logs in to the VMware Cloud Director API, the `OrgList` in the returned `Session` element contains a link to the `System` organization.

```
<OrgList ... >
   ...
   <Org
      type="application/vnd.vmware.admin.systemOrganization+xml"
      name="System"
      href="https://vcloud.example.com/api/admin/org/123"/>
   ...
</OrgList>
```

Organization Administration

System administrators create organizations and organization administrators, and establish certain organization policies. Organization administrators populate their organization with users and groups, create and assign roles, and can update most organization policies and properties.

A cloud can contain one or more organizations. Each organization is a unit of administration for a collection of users, groups, and computing resources. Users authenticate at the organization level, supplying credentials established when the user was created or imported. User credentials are authenticated by the organization's identity provider. VMware Cloud Director includes an integrated identity provider. It also supports several standards-based external identity providers.

Retrieve or Update Organization Settings

Organization settings define organization policies such as default lease settings for vApps and how incorrect login attempts are handled. They also configure how the organization uses services such as email, LDAP, and identity providers.

An `AdminOrg` element contains an `OrgSettings` element, which contains the following elements, each of which represents a group of related organization settings. Default settings are inherited from the system.

`GeneralOrgSettings`

Specifies storage and deployment quotas and other behaviors for virtual machines that members of this organization own. Sets the scope of catalog publication and subscription in this organization.

`VAppLeaseSettings`

Controls storage and deployment leases for vApps.

`VAppTemplateLeaseSettings`

Controls storage and deployment leases for vApp templates.

`OrgLdapSettings`

Defines whether this organization is connected to an LDAP service, and whether it uses the service defined in the system `LdapSettings` or a custom LDAP service. See Configuring and Managing Federation with LDAP.

`OrgOAuthSettings`

Defines the OAuth identity provider used by this organization. See Configuring and Managing Federation with OAuthConfiguring and Managing Federation with OAuth.

`OrgEmailSettings`

Defines whether this organization uses the email service defined in the system `EmailSettings` or a custom email service.

`OrgPasswordPolicySettings`

Specifies policies to be followed when a user in this organization enters an incorrect password. Initial values are inherited from the system `PasswordPolicySettings`.

`OrgOperationLimitsSettings`

Specifies limits to be placed on simultaneous resource-intensive operations and console sessions for members of this organization.

`OrgGuestPersonalizationSettings`

Default values for `GuestCustomizationSection` elements in virtual machines created by this organization. See Retrieve or Modify the GuestCustomizationSection of a Virtual Machine

`OrgFederationSettings`

Settings related to any SAML identity provider that this organization shares with other applications or enterprises to enable single sign-on. See Configuring and Managing Federation with SAML.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Retrieve the list of organization settings elements.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26/settings
```

The response is an `OrgSettings` element.

2 Examine the `OrgSettings` element to find the links to the sections to view or modify.

Each section is represented in the `OrgSettings` element with a link where `rel="down"`. You can use that link to retrieve the section. The retrieved section includes a link where `rel="edit"`. You can use that link as the target of a PUT request that modifies the settings that the element represents. The `OrgSettings` element itself also has a `rel="edit"` link, which you can use to update multiple settings sections in one request.

3 Retrieve the settings element to modify.

Make a GET request to the URL in the element's `href` attribute value.

4 Modify the retrieved settings element.

Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. `Link` elements and `href` attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

5 Update the settings with your modifications.

Find the `Link` element in the settings element where `rel="edit"`. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body. See the request portion of Update Organization General Settings.

Example: Update Organization General Settings

This example updates the general settings of the organization created in Create an Organization. When you create or retrieve an `AdminOrg`, these settings are contained in the `OrgGeneralSettings` element. To update them, you must use a `GeneralOrgSettings` element, which has the same contents as `OrgGeneralSettings`. This update changes the limits on deployed and stored virtual machines. The request includes all members of the `GeneralOrgSettings` element, even those that are not changing. It is a best practice to include all members of the `GeneralOrgSettings` element in an update request. Optional elements that are missing or empty in the request are reset to their default values.

Request:

```
PUT https://vcloud.example.com/api/admin/org/26/settings/general
Content-Type: application/vnd.vmware.admin.organizationGeneralSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<GeneralOrgSettings
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.organizationGeneralSettings+xml">
   <CanPublishCatalogs>false</CanPublishCatalogs>
   <CanPublishExternally>true</CanPublishExternally>
   <CanSubscribe>false</CanSubscribe>
   <DeployedVMQuota>10</DeployedVMQuota>
   <StoredVmQuota>100</StoredVmQuota>
   <UseServerBootSequence>false</UseServerBootSequence>
   <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
</GeneralOrgSettings>
```

The response contains information extracted from the request, and includes a `rel="edit"` link and other attributes that the server creates.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.organizationGeneralSettings+xml
...
<GeneralOrgSettings
   type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
   href="https://vcloud.example.com/api/admin/org/26/settings/general">
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
      href="https://vcloud.example.com/api/admin/org/26/settings/general" />
   <CanPublishCatalogs>false</CanPublishCatalogs>
   <CanPublishExternally>true</CanPublishExternally>
   <CanSubscribe>false</CanSubscribe>
   <DeployedVMQuota>10</DeployedVMQuota>
   <StoredVmQuota>100</StoredVmQuota>
   <UseServerBootSequence>false</UseServerBootSequence>
   <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
</GeneralOrgSettings>
```

When you retrieve the organization after updating its `GeneralOrgSettings`, you can see the results of the update in the `OrgGeneralSettings` element of the `AdminOrg`.

```
GET https://vcloud.example.com/api/admin/org/26
...
<AdminOrg ...>
   ...
      <OrgGeneralSettings
         type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
         href="https://vcloud.example.com/api/admin/org/26/settings/general">
         <Link
            rel="edit"
            type="application/vnd.vmware.admin.organizationGeneralSettings+xml"
            href="https://vcloud.example.com/api/admin/org/26/settings/general" />
         <CanPublishCatalogs>false</CanPublishCatalogs>
         <CanPublishExternally>true</CanPublishExternally>
         <CanSubscribe>false</CanSubscribe>
         <DeployedVMQuota>10</DeployedVMQuota>
         <StoredVmQuota>100</StoredVmQuota>
         <UseServerBootSequence>false</UseServerBootSequence>
         <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
      </OrgGeneralSettings>
   ...
 </AdminOrg>
```

VDC Administration

A newly created organization has no VDCs in it. A system administrator can use system resources to create VDCs in an organization. A system administrator can also define VDC templates, share them with organizations and grant organization members the right to use the templates to create VDCs in their organization.

An organization virtual datacenter (organization VDC) is a deployment environment for virtual systems owned by the containing organization, and an allocation mechanism for resources such as networks, storage, CPU, and memory. In an organization VDC, computing resources are fully virtualized, and can be allocated based on demand, service level requirements, or a combination of the two.

Create a VDC from a Template

A system administrator can define VDC templates, share them with organizations, and grant organization members the right to use the templates to create VDCs in their organization.

A VDC template specifies a VDC configuration. If the configuration includes an Edge Gateway, the VDC can support creation of routed organization VDC networks.

Note If you are a system administrator, you can create a VDC without using a template. See Add a VDC to an Organization .

Prerequisites

This operation requires the rights included in the predefined **Organization Administrator** role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the admin view of your organization.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26
```

2 Examine the response to locate the Link element that contains the URL for retrieving the list of VDC templates that are available to your organization.

This element has a rel attribute value of down and a type attribute value of application/vnd.vmware.admin.vdcTemplates+xml, as shown here:

```
<Link
   rel="down"
   href="https://vcloud.example.com/api/vdcTemplates/"
   type="application/vnd.vmware.admin.vdcTemplates+xml"/>
```

3 Retrieve the list of VDC templates.

Make a GET request to the href value in the link shown in Step 2. The response is a VdcTemplateList that contains one or more VdcTemplate elements. Each of these elements has an href attribute whose value is a URL you can GET to retrieve the representation of the template. If there are no VDC templates in the list, you can ask your system administrator to create a template and share it with your organization.

4 Retrieve a template from the list.

Each VdcTemplate in the list includes a Description that provides more information about the features of the VDC that the template creates. For additional information, you can retrieve the VdcTemplateSpecification from the VdcTemplate. All of these elements are read-only.

5 Instantiate the template to create a VDC in your organization.

Each organization includes an action/instantiate link that you can use with a POST request to add a VDC to your organization that a template specifies. See Create a VDC From a Template. The system administrator can impose limits on the number of VDCs that any organization can create. If your organization already contains the maximum number of VDCs allowed by the system administrator, the request fails.

Example: Create a VDC From a Template

This request creates a VDC from a template. If you do not specify the `name` attribute or include a `Description` element, the new VDC is created with the `name` and `Description` shown in the template.

Request:

```
POST https://vcloud.example.com/api/org/26/action/instantiate
Content-Type: application/vnd.vmware.vcloud.instantiateVdcTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateVdcTemplateParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   Name="templateVdc">
   <Description>Example VDC from template</Description>
   <Source
      href="https://vcloud.example.com/api/VdcTemplate/100"/>
</InstantiateVdcTemplateParams>
```

The response is a `Task`.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
<Task
   name="task"
   status="running"
   operation="Creating Virtual Datacenter templateVdc (100)" ...
   ...
</Task>
```

Change the Name or Description of an Existing VDC

An administrator can update an existing virtual datacenter to change its name or description.

Prerequisites

- This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.
- Retrieve the VDC whose name or description you want to change. If you don't know the URL of the VDC, you can use a query like this one to retrieve a list of references all VDCs in your organization.

  ```
  GET https://vcloud.example.com/api/query?type=adminOrgVdc&format=references
  ```

Procedure

1 Examine the `Vdc` to find its `rel="edit"` link.

2 Create a `Vdc` element that contains the new name and description.

   This `Vdc` element need only specify the new `name` and `Description`. See Change the Name and Description of an Existing VDC.

3 Make a PUT request to the `rel="edit"` link to change the name or description of the VDC .

   Supply the `Vdc` element as the request body.

Example: Change the Name and Description of an Existing VDC

This example changes the name and description of the VDC created in Create a VDC From a Template.

Request:

```
PUT https://vcloud.example.com/api/vdc/130
Content-Type: application/vnd.vmware.vcloud.vdc+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Vdc
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="NewVdcName"
   type="application/vnd.vmware.vcloud.vdc+xml">
   <Description>New VDC description</Description>
</Vdc>
```

The response is a `Task`.

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
<Task
   name="task"
   status="running"
   operation="Updating Virtual Datacenter templateVdc (130)" ...
   ...
</Task>
```

Update Organization VDC Storage Profiles

A system administrator can update the storage profiles that are available in an organization VDC. You can add new storage profiles and remove unused storage profiles.

An organization VDC storage profile allocates a subset of the storage available in a Provider VDC storage profile for use by vApp templates, virtual machines, and media objects in the organization VDC. For each organization VDC storage profile you create, you must specify a storage limit, which cannot exceed the storage available in the Provider VDC storage profile (the value of `CapacityTotal`–`CapacityUsed` in the `ProviderVdcStorageProfile`). When you update organization VDC storage profiles, you can change the default storage profile and modify the limits on existing storage profiles.

Note Storage profiles are represented as Storage Policies in the VMware Cloud Director HTML5 UI.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the VDC in the admin view.

  Use a request like this one:

```
GET https://vcloud.example.com/api/admin/vdc/44
```

2 Examine the `AdminVdc` element to find the `vdcStorageProfiles` link, `VdcStorageProfiles` element, and `ProviderVdcReference` element it contains.

  The `vdcStorageProfiles` link has the following form:

```
<Link
   rel="edit"
   type="application/vnd.vmware.admin.updateVdcStorageProfiles+xml"
   href="https://vcloud.example.com/api/admin/vdc/44/vdcStorageProfiles" />
```

3 Create an `UpdateVdcStorageProfiles` request body that specifies the details of the update.

- To add a storage profile:

  a Select a storage profile from the Provider VDC referenced in the `ProviderVdcReference` element of the VDC you are updating.

    The storage profile must not be listed in the `VdcStorageProfiles` element of the VDC you are updating.

  b Include an `AddStorageProfile` element in the `UpdateVdcStorageProfiles` request body.

    The `AddStorageProfile` element must specify values for `Units`, `Limit`, and `Default`, and must include a reference to the Provider VDC storage profile on which it is based. You can add multiple storage profiles in a single request. Only one of them can specify `Default` as `true`. If any `AddStorageProfile` element specifies `Default` as `true`, that storage profile becomes the new default storage profile for the VDC.

- To remove a storage profile:

  a Examine the VdcStorageProfiles element and find the profile to remove.

  b Verify that it is not the default storage profile for the VDC, and that no virtual machines are using it.

  You can use the adminVm query and filter on the storageProfileName attribute to list all storage profiles that are in use.

  c Create an UpdateVdcStorageProfiles element that contains a RemoveStorageProfile element for each storage profile to remove.

4 POST the UpdateVdcStorageProfiles element to the VDC's vdcStorageProfiles link.

Example: Update VDC Storage Profiles

This request adds a storage profile to the VDC created in Create an Organization VDC with Pay As You Go Reservation Allocation Model. The new storage profile is one of the profiles available from the Provider VDC that backs this organization VDC.

One way to retrieve a list of all the Provider VDC storage profiles available from a specific Provider VDC is to use the query service. This query applies a filter that selects only those storage profiles available from the Provider VDC that backs the organization VDC created in Create an Organization VDC with Pay As You Go Reservation Allocation Model.

```
GET https://vcloud.example.com/api/query?type=providerVdcStorageProfile&format=references
&filter=providerVdc==https://vcloud.example.com/api/admin/providervdc/35
```

The response might look something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderVdcStorageProfileReferences .../>
   ...
   <ProviderVdcStorageProfileReference
      type="application/vnd.vmware.admin.pvdcStorageProfile+xml"
      name="Gold"
      id="urn:vcloud:providervdcstorageprofile:101"
      href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101" />
   <ProviderVdcStorageProfileReference
      type="application/vnd.vmware.admin.pvdcStorageProfile+xml"
      name="Silver"
      id="urn:vcloud:providervdcstorageprofile:128"
      href="https://vcloud.example.com/api/admin/pvdcStorageProfile/128" />
   ...
</ProviderVdcStorageProfileReferences>
```

You can use the information in the response to construct the AddStorageProfile element in the request body. This example creates a storage profile that is not a default storage profile, and has a specific value for Limit, 5038 MB. To specify unlimited storage (subject to the capacity of the underlying Provider VDC), set the value of Limit to 0.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/vdcStorageProfiles
Content-Type: application/vnd.vmware.admin.updateVdcStorageProfiles+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<UpdateVdcStorageProfiles
   xmlns="http://www.vmware.com/vcloud/v1.5" >
   <AddStorageProfile>
      <Enabled>true</Enabled>
      <Units>MB</Units>
      <Limit>5038</Limit>
      <Default>false</Default>
      <ProviderVdcStorageProfile
         href="https://vcloud.example.com/api/admin/pvdcStorageProfile/128" />
   </AddStorageProfile>
</UpdateVdcStorageProfiles>
```

The response is a `Task`.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
   ...
</Task>
```

Enable, Disable, or Remove a VDC

A system administrator can use links in an `AdminVdc` element to enable, disable, or remove an organization VDC.

Prerequisites

- This operation is restricted to system administrators.
- Retrieve the VDC that you want to enable, disable, or remove. If you don't know the URL of the VDC, you can use a query like this one to retrieve a list of references all VDCs in the system.

  ```
  GET https://vcloud.example.com/api/query?type=adminOrgVdc&format=references
  ```

Procedure

- To enable a VDC, POST a request to its `action/enable` link.
- To disable a VDC, POST a request to its `action/disable` link.

  When you disable an organization VDC, you prevent further use of its compute and storage resources. Running vApps and powered on virtual machines continue to run, but you cannot create or start additional vApps or virtual machines.

- To remove a VDC, remove all the objects it contains, then disable and remove it.

  a Relocate or remove any vApps that have been deployed in the VDC.

  b Remove any organization VDC networks that the VDC contains.

  c Remove any Edge Gateways that the VDC contains.

  d Disable the VDC by making POST a request to its `action/disable` link.

  After the VDC is disabled, its representation includes a `rel="remove"` link if it no longer contains any objects.

  e Make a DELETE request to the VDC's `rel="remove"` link.

Note You can make a request like this one, which adds the query string `recursive=true` to the VDC href, to remove a VDC that contains one or more objects as long as those objects are in a state that normally allows removal.

```
DELETE https://vcloud.example.com/api/admin/vdc/44?recursive=true
```

You can use an additional query parameter to force this kind of recursive removal even when the VDC contains objects that are not in an appropriate state.

```
DELETE https://vcloud.example.com/api/admin/vdc/44?recursive=true&force=true
```

Results

The server takes the requested action and returns an HTTP status of 204 No Content.

Apply Access Controls to a VDC

Upon creation, an organization VDC grants full access to all members of the containing organization. An administrator can use the VMware Cloud Director API access control mechanism to restrict access to specific users.

Organization VDCs implement a subset of the access control features described in Controlling Access to vApps and Catalogs. To restrict access to a VDC, you first apply access controls that deny use of the VDC to all users. After you do that, you can make exceptions to grant access to up to 128 individual users. You apply VDC access controls using a `controlAccess` request and `ControlAccessParams` request body. Values of certain elements in the request body have special meanings when applied to a VDC.

IsSharedToEveryone

The value of this element specifies whether the VDC imposes any access controls. If it is set to `false`, access is denied to all users except the ones references in the `AccessSettings` element. If it is set to `true`, no access controls apply even if you have defined them in `AccessSettings`.

AccessLevel

A value of `ReadOnly` grants the subject all rights to use the VDC. In this release, `ReadOnly` is the only legal VDC `AccessLevel` for a user.

Prerequisites

This operation requires the rights included in the predefined **Organization Administrator** role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the VDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/vdc/130
```

2 Examine the `AdminVdc` element to find the `controlAccess` links that it contains.

3 Create a `ControlAccessParams` element request body that specifies the details of the update.

See Apply Access Controls to a VDC.

4 PUT the `ControlAccessParams` element to the `action/controlAccess` link that you retrieved in Step 2.

Example: Apply Access Controls to a VDC

This request updates the access controls of a VDC to grant access to two external users defined in an OAuth identity provider.. The request body, a `ControlAccessParams` element, specifies a value of `false` for the `IsSharedToEveryone` element, which denies access to all users. It also includes an `AccessSetting` element for each user to whom access is granted. Each of these users is identified by an `ExternalSubject` element. An `ExternalSubject` identifies a user account defined in a supported OAuth or SAML identity provider. See About Federation and Single Sign-On. In this element, the `SubjectId` is the user name with which the user logs in to the identity provider whose type is specified in `IdpType`. The user must be a member of the organization that owns the VDC.

Request:

```
PUT https://vcloud.example.com/api/vdc/130/action/controlAccess
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ControlAccessParams xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      <AccessSetting>
         <ExternalSubject>
            <SubjectId>user1@example.com</SubjectId>
            <IsUser>true</IsUser>
            <IdpType>OAUTH</IdpType>
         </ExternalSubject>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
      <AccessSetting>
         <ExternalSubject>
            <SubjectId>user2@example.com</SubjectId>
            <IsUser>true</IsUser>
            <IdpType>OAUTH</IdpType>
         </ExternalSubject>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
   </AccessSettings>
</ControlAccessParams>
```

A user defined in the integrated identity provider is not considered external. To specify users who are defined by the integrated identity provider, use `Subject`, not `ExternalSubject`, as shown in this fragment.

```
<ControlAccessParams xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      <AccessSetting>
        <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/45"/>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
    </AccessSettings>
</ControlAccessParams>
```

The response, a subset of which appears here, echoes the request.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      ...
   </AccessSettings>
</ControlAccessParams>
```

Network Administration

A newly created organization VDC has no networks in it. After a system administrator has created the required network infrastructure, an organization administrator can create and manage most types of organization VDC networks.

An organization VDC can be provisioned with zero or more networks. These organization VDC networks can be configured to provide direct or routed connections to external networks, or can be isolated from external networks and other organization VDC networks. Routed connections require an Edge Gateway and network pool in the VDC. The Edge Gateway provides firewall, network address translation, static routing, VPN, and load balancing services.

In addition to the network operations enabled by the VMware Cloud Director API, you can use the VMware Cloud Director API for NSX for advanced networking operations. See the *VMware Cloud Director API for NSX Programming Guide*.

About VMware Cloud Director Networks

There are three categories of VMware Cloud Director networks: external networks, organization VDC networks, and vApp networks. Additional infrastructure objects such as Edge Gateways and network pools are required by most categories of networks and must be created by a system administrator.

You must be a system administrator to create an external network, a directly connected organization VDC network, a network pool, or an Edge Gateway. An organization administrator can create and modify routed and isolated organization VDC networks, and any user who has vApp Author rights can create and modify a vApp network.

vApp Networks

A vApp network is a logical network that controls how the virtual machines in a vApp connect to each other and to organization VDC networks. Users can create and update vApp networks and connect them to organization VDC networks. See About vApp Networks.

Organization VDC Networks

An organization VDC network allows virtual machines in the organization VDC to communicate with each other and to access other networks, including organization VDC networks and external networks, either directly or through an Edge Gateway that can provide firewall and NAT services.

- A direct organization VDC network connects directly to an eternal network. Can be IPv4 or IPv6.

  Only a system administrator can create a direct organization VDC network.

- A routed organization VDC network connects to an external network through an Edge Gateway. A routed organization VDC network also requires the containing VDC to include a network pool. After a system administrator has provisioned an organization VDC with an Edge Gateway and associated it with a network pool, organization administrator or system administrators can create routed organization VDC networks in that VDC. Can be IPv4 or IPv6.

- An isolated organization VDC network does not require an Edge Gateway or external network. Provides an isolated, private network that machines in the organization VDC can connect to. Can be backed by either a network pool or an NSX-T logical switch. Can be IPv4 only.

After a system administrator has created an organization VDC with a network pool, organization administrators or system administrators can create isolated organization VDC networks in that VDC.

Only the system administrator can create and manage NSX-T organization virtual data center networks by using theVMware Cloud Director OpenAPI or the VMware Cloud Director Service Provider Admin Portal.

- A cross-VDC network is part of a stretched network spanning a data center group. Can be IPv4 only.

  Only the system administrator can create and manage cross-VDC networks by using the VMware Cloud Director OpenAPI or the VMware Cloud Director Tenant Portal.

For information about using the VMware Cloud Director OpenAPI, see *Getting Started with VMware Cloud Director OpenAPI* at https://code.vmware.com.

Most types of organization VDC networks do not provide any network services. Isolated organization VDC networks can specify a `DhcpPoolService`, which provides DHCP addresses from several pools of IP address ranges. All other services, such as NAT, firewall, and load balancing, are configured by a system administrator on the Edge Gateway to which the network connects.

By default, only virtual machines in the organization VDC that contains the network can use it. When you create an organization VDC network, you can specify that it is shared. A shared organization VDC network can be used by all virtual machines in the organization.

Edge Gateways

An Edge Gateway is a virtual router for organization VDC networks. You must be a system administrator to create an Edge Gateway.

VMware Cloud Director supports IPv4 and IPv6 Edge Gateways.

An Edge Gateway can provide any of the following services, defined in the `GatewayFeatures` element of the Edge Gateway's `Configuration`.

`FirewallService`

Specifies firewall rules that, when matched, block or allow incoming or outgoing network traffic. See Firewall Service Configurations.

`GatewayDhcpService`

Provides DHCP services to virtual machines on the network. A variant of this service, `DhcpService`, is intended to provide DHCP services in vApp networks. See Gateway DHCP Service Configurations.

`GatewayIpsecVpnService`

Defines one or more virtual private networks that connect an Edge Gateway to another network in or outside of the cloud.

`LoadBalancerService`

Distributes incoming requests across a set of servers. See Load Balancer Service Configurations.

`NatService`

Provides network address translation services to computers on the network.

`StaticRoutingService`

Specifies static routes to other networks. See Static Routing Service Configurations.

For an example of adding services to an Edge Gateway, see Configure Edge Gateway Services. For more information about any of these services, see the *vShield Administration Guide*.

Important IPv6 edge gateways support limited services. In VMware Cloud Director, IPv6 edge gateways support edge firewalls, distribute firewalls, and static routing.

External Networks and Network Pools

External networks and network pools are vSphere resources backed by vSphere portgroup, VLAN, or DVswitch objects. A system administrator must create them, as described in Create an External Network and Create a Network Pool. As a system administrator, you must supply a reference to an external network when you create an Edge Gateway. An organization VDC must include a reference to a network pool or it will not be able to able to contain routed or isolated networks. See Retrieve a List of External Networks and Network Pools

Configure Edge Gateway Services

An administrator can configure NAT, firewall, and similar services on an existing Edge Gateway by updating its `EdgeGatewayServiceConfiguration`.

The `Configuration` element of an `EdgeGateway` includes an `EdgeGatewayServiceConfiguration` element, which can contain definitions of any of the services listed in Edge Gateways. Details of service configurations vary, but the mechanism is the same for creating or updating any Edge Gateway service. Note that some services require a reference to one or more Edge Gateway interfaces, and cannot be configured until those interfaces exist.

Important IPv6 edge gateways support limited services. In VMware Cloud Director, IPv6 edge gateways support edge firewalls, distribute firewalls, and static routing.

Prerequisites

This operation requires the rights included in the predefined **Organization Administrator** role or an equivalent set of rights.

Verify that your organization VDC contains an Edge Gateway. If it does not, a system administrator can create one.

Verify that the Edge Gateway is not an Advanced Gateway. If the `EdgeGateway` element that represents this Edge Gateway has an `AdvancedNetworkingEnabled` element whose value is `true`, using the VMware Cloud Director API to configure Edge Gateway services can produce unexpected results. Use the VMware Cloud Director API for NSX instead. See VMware Knowledge Base article http://kb.vmware.com/kb/2147625

Procedure

1. Retrieve the XML representation of the Edge Gateway.
2. Examine the response to locate the `Link` element that contains the URL for configuring services on the Edge Gateway.

   This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.edgeGatewayServiceConfiguration+xml`, as the following example shows:

   ```
   <Link
      href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/configureServices"
      rel="add"
      type="application/vnd.vmware.admin.edgeGatewayServiceConfiguration+xml"/>
   ```
3. Copy the `EdgeGatewayServiceConfiguration` element from the `EdgeGateway` you retrieved in Step 1.

   The `configureServices` action replaces the entire contents of the existing `EdgeGatewayServiceConfiguration` with the one in the request body. Using the existing `EdgeGatewayServiceConfiguration` as the basis for your modifications reduces the chances of unintentional service changes.
4. Modify the `EdgeGatewayServiceConfiguration` that you copied in Step 3 to add, remove, or change the services that this Edge Gateway offers.

   An `EdgeGatewayServiceConfiguration` element can contain any of the following elements:

   - `FirewallService`
   - `GatewayDhcpService`
   - `GatewayIpsecVpnService`
   - `LoadBalancerService`
   - `NatService`
   - `StaticRoutingService`
5. POST the modified `EdgeGatewayServiceConfiguration` element to the URL in the value of the `href` attribute of the `configureServices` link described in Step 2.

Results

The server takes the requested action and returns a `Task` element that tracks the progress of the request.

When the task completes successfully, the `EdgeGatewayServiceConfiguration` element you POSTed replaces the one you copied in Step 3.

Example: Configure Services on an Edge Gateway

This example replaces the default firewall service on the Edge Gateway created in Create an Edge Gateway. For details about this `FirewallService`, see Firewall Service Configurations

Request:

```
POST https://vcloud.example.com/api/admin/edgeGateway/2000/action/configureServices
Content-Type: application/vnd.vmware.admin.edgeGatewayServiceConfiguration+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<EdgeGatewayServiceConfiguration
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <FirewallService>
      <IsEnabled>true</IsEnabled>
      <DefaultAction>allow</DefaultAction>
      <LogDefaultAction>false</LogDefaultAction>
      <FirewallRule>
         <IsEnabled>true</IsEnabled>
         <Description>allow incoming ssh</Description>
         <Policy>allow</Policy>
         <Protocols>
            <Tcp>true</Tcp>
         </Protocols>
         <DestinationPortRange>22</DestinationPortRange>
         <DestinationIp>Internal</DestinationIp>
         <SourcePortRange>Any</SourcePortRange>
         <SourceIp>External</SourceIp>
         <EnableLogging>true</EnableLogging>
      </FirewallRule>
      <FirewallRule>
         <IsEnabled>true</IsEnabled>
         <Description>deny incoming telnet</Description>
         <Policy>drop</Policy>
         <Protocols>
            <Tcp>true</Tcp>
         </Protocols>
         <DestinationPortRange>23</DestinationPortRange>
         <DestinationIp>Internal</DestinationIp>
         <SourcePortRange>Any</SourcePortRange>
         <SourceIp>External</SourceIp>
         <EnableLogging>false</EnableLogging>
      </FirewallRule>
   </FirewallService>
</EdgeGatewayServiceConfiguration>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
   href="https://vcloud.example.com/api/task/2120"
   ...
   status="running"
   operation="Updating services EdgeGateway theEdge(2000)"
   ... >
</Task>
```

Firewall Service Configurations

The default `FirewallService` in an `EdgeGatewayServiceConfiguration` is enabled and configured to block all incoming traffic. You can modify that `FirewallService` to allow incoming traffic, block outgoing traffic, or both.

A firewall service configuration includes several important parameters.

Firewall Rules

Each firewall rule specifies a protocol, IP address, and port. Packets that match the criteria in the rule are subject to an action defined in the `Policy` element of the rule. The action can forward the packet to the destination IP address and port, or drop it and optionally log a message describing the packet that was dropped. Packets that do not match any rule are subject to the policy contained in the `DefaultAction` element of the `FirewallService`.

Firewall Rule Logging

Firewall rule actions can be logged to the system syslog server, and optionally to a syslog server you create and manage. When you specify a value of `true` in the `EnableLogging` element of a `FirewallRule`, all packets that trigger the rule are logged to the system `syslog` server. Logging for all rules is controlled by the value of the `LogDefaultAction` element of the `FirewallService`.

To log firewall rule messages from this Edge Gateway to your own `syslog` server in addition to the system `syslog` server, add a `SyslogServerSettings` element to its `Configuration` element and specify your `syslog` server's IP address in the `SyslogServerIp` element of its `TenantSyslogServerSettings`.

Port and Address Ranges

These elements in a `FirewallRule` specify source and destination IP ports and addresses to which the rule applies.

Example: Firewall Service Definition with Two Rules

This fragment of an `EdgeGatewayServiceConfiguration` defines a firewall service with two rules: one that allows incoming SSH connection, and one that denies incoming Telnet connections. These rules apply to any virtual machine that connects to a network backed by this Edge Gateway. Each rule is defined in a `FirewallRule` element, and can include the following specifications:

Policy

The default policy value, `allow`, causes the firewall to forward packets that match the rules. Specify `drop` to drop packets that match the rules.

Protocols

By default, a rule applies to both UDP and TCP protocols. You can limit the rule to one protocol or the other by including `Tcp` and `Udp` elements in `Protocols` and specifying a value of `true` or `false` for each.

SourcePortRange

Specify a source IP port or port range, or set to `any` to match any port.

DestinationPortRange

Specify a destination IP port or port range, or set to `any` to match any port.

SourceIp

Specify a source IP address, or use one of these strings.

Table 7-1. SourceIp and DestinationIp Values

Value	Result
Any	Matches any IP address
Internal	Matches any IP address originating on an organization VDC network connected to this EdgeGateway. When used in a vApp network, matches any IP address assigned to a virtual machine in the vApp.
External	Matches any IP address originating on an external network connected to this EdgeGateway. When used in a vApp network, matches any IP address except those assigned to a virtual machine in the vApp.

DestinationIp

Specify a source IP address, or use one of the strings shown in Table 7-1. SourceIp and DestinationIp Values.

EnableLogging

Set to `true` to log all packets that trigger this rule. See Firewall Rule Logging.

Rules are applied to packets in the order in which the `FirewallRule` elements appear in the `FirewallService` definition.

Note The system assigns an `Id` value to each rule you create and uses these values when logging rule actions.

```
<FirewallService>
   <IsEnabled>true</IsEnabled>
   <DefaultAction>allow</DefaultAction>
   <LogDefaultAction>false</LogDefaultAction>
   <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>allow incoming ssh</Description>
      <Policy>allow</Policy>
      <Protocols>
         <Tcp>true</Tcp>
      </Protocols>
      <DestinationPortRange>22</DestinationPortRange>
      <DestinationIp>Internal</DestinationIp>
      <SourcePortRange>Any</SourcePortRange>
      <SourceIp>External</SourceIp>
      <EnableLogging>false</EnableLogging>
   </FirewallRule>
   <FirewallRule>
      <IsEnabled>true</IsEnabled>
      <Description>deny incoming telnet</Description>
      <Policy>drop</Policy>
      <Protocols>
         <Tcp>true</Tcp>
      </Protocols>
      <DestinationPortRange>23</DestinationPortRange>
      <DestinationIp>Internal</DestinationIp>
      <SourcePortRange>Any</SourcePortRange>
      <SourceIp>External</SourceIp>
      <EnableLogging>false</EnableLogging>
   </FirewallRule>
</FirewallService>
```

You can see this fragment in the context of an Edge Gateway configuration in Configure Services on an Edge Gateway.

NAT Service Configurations

An Edge Gateway configuration can define a NAT (Network Address Translation) service that translates source or destination IP addresses and port numbers. In the most common case, you associate a NAT service with an uplink interface on an Edge Gateway so that addresses on organization VDC networks are not exposed on the external network.

A NAT service in an `EdgeGatewayServiceConfiguration` can include one or more rules, each of which is expressed in a `GatewayNatRule` element. Each rule translates the original IP address, port, or both, and applies to a network connected to the Edge Gateway. If the network is an uplink (to an external network), the network must include an IP sub-allocation pool.

There are two kinds of rules, as expressed in the value of the RuleType element:

SNAT

Source network address translation. This kind of rule translates the packet's source address and, optionally, source IP port to the values you specify.

DNAT

Destination network address translation. This kind of rule translates the packet's destination address and, optionally, destination IP port to the values you specify.

Example: NAT Service

The following fragment of an EdgeGatewayServiceConfiguration defines and enables a NatService that applies one destination NAT rule and one source NAT rule to the uplink interface defined in Create an Edge Gateway. In the DNAT rule, the OriginalIp and OriginalPort apply to the destination IP address and port of the packet being inspected. In the SNAT rule, the OriginalIp and OriginalPort apply to the source IP address and port of the packet being inspected. When you create an SNAT rule, you do not need to specify values for TranslatedPort and OriginalPort, which default to any.

Note The system assigns an Id value to each rule you create and uses these values when logging rule actions.

```
<?xml version="1.0" encoding="UTF-8"?>
<NatService>
   <IsEnabled>true</IsEnabled>
   <NatRule>
      <RuleType>DNAT</RuleType>
      <IsEnabled>true</IsEnabled>
      <GatewayNatRule>
         <Interface
            href="https://vcloud.example.com/api/admin/network/297" />
         <OriginalIp>10.147.115.155</OriginalIp>
         <OriginalPort>any</OriginalPort>
         <TranslatedIp>192.168.0.10</TranslatedIp>
         <TranslatedPort>any</TranslatedPort>
         <Protocol>any</Protocol>
         <IcmpSubType>any</IcmpSubType>
      </GatewayNatRule>
   </NatRule>
   <NatRule>
      <RuleType>SNAT</RuleType>
      <IsEnabled>true</IsEnabled>
      <GatewayNatRule>
         <Interface
            href="https://vcloud.example.com/api/admin/network/297" />
         <OriginalIp>192.168.0.10-192.168.0.255</OriginalIp>
         <TranslatedIp>10.147.115.155</TranslatedIp>
         <Protocol>any</Protocol>
      </GatewayNatRule>
   </NatRule>
</NatService>
```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See Configure Services on an Edge Gateway.

Static Routing Service Configurations

An Edge Gateway or routed vApp network configuration can define a static routing service that specifies one or more static routes.

You can create static routes between two organization VDC networks, or between routed vApp networks that do not have overlapping IP address spaces. Static routing service details and routes are defined in a `StaticRoutingService` element contained by the `Features` element of a vApp network's `Configuration` or the `GatewayFeatures` element of an Edge Gateway's `Configuration`. A `StaticRoutingService` element can contain zero or more `StaticRoute` elements. Each `StaticRoute` specification requires the following elements.

Name

Name for the route.

Network

Network specification in CIDR notation.

NextHopIp

IP address of the next hop on the route. This address is typically the value in the `ExternalIp` element of the `RouterInfo` from the network to which this static route connects.

Interface

Specify `internal` if `NextHopIp` contains an IP address in the same network. Specify `external` if `NextHopIp` contains an IP address in a different network.

GatewayInterface

Used when configuring a static route in an organization VDC network. Contains a reference to the organization VDC network for which the static route is configured.

Example: Static Routes Between Organization VDC Networks

Assume two routed organization VDC networks, as defined in this fragment of an `EdgeGateway` element.

```
<GatewayInterface>
   <Name>vnic2</Name>
   <DisplayName>routed1</DisplayName>
   <Network
      type="application/vnd.vmware.admin.network+xml"
      name="routed1"
      href="https://vcloud.example.com/api/admin/network/29" />
   <InterfaceType>internal</InterfaceType>
   <SubnetParticipation>
      <Gateway>192.168.3.1</Gateway>
      <Netmask>255.255.255.0</Netmask>
      <IpAddress>192.168.3.1</IpAddress>
   </SubnetParticipation>
   <IpRanges />
   <UseForDefaultRoute>false</UseForDefaultRoute>
</GatewayInterface>
<GatewayInterface>
   <Name>vnic3</Name>
   <DisplayName>routed2</DisplayName>
   <Network
      type="application/vnd.vmware.admin.network+xml"
      name="routed2"
      href="https://vcloud.example.com/api/admin/network/67" />
   <InterfaceType>internal</InterfaceType>
   <SubnetParticipation>
      <Gateway>172.168.0.1</Gateway>
      <Netmask>255.255.255.0</Netmask>
      <IpAddress>172.168.0.1</IpAddress>
   </SubnetParticipation>
   <IpRanges />
   <UseForDefaultRoute>false</UseForDefaultRoute>
</GatewayInterface>
```

- The organization VDC network named `routed1` has `Gateway` address `192.168.3.1`.
- The organization VDC network named `routed2` has `Gateway` address `172.168.0.1`.
- The `Configuration` of the vApp network in vApp1 has a `RouterInfo` element whose `ExternalIp` value is `192.168.2.100`.
- The `Configuration` of the vApp network in vApp2 has a `RouterInfo` element whose `ExternalIp` value is `192.168.1.100`.

A `StaticRoutingService` defined by the following fragment creates a static route to vApp1 from network `routed1`, and a static route to vApp2 from network `routed2`.

```
<StaticRoutingService>
   <IsEnabled>true</IsEnabled>
   <StaticRoute>
      <Name>TovApp1</Name>
      <Network>192.168.2.0/24</Network>
      <NextHopIp>192.168.3.10</NextHopIp>
      <Interface>Internal</Interface>
      <GatewayInterface
         type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
         name="routed1"
         href="https://vcloud.example.com/api/admin/network/29" />
   </StaticRoute>
   <StaticRoute>
      <Name>TovApp2</Name>
      <Network>192.168.1.0/24</Network>
      <NextHopIp>172.168.0.10</NextHopIp>
      <Interface>Internal</Interface>
      <GatewayInterface
         type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
         name="routed2"
         href="https://vcloud.example.com/api/admin/network/67" />
   </StaticRoute>
</StaticRoutingService>
```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See Configure Services on an Edge Gateway.

Static Routes Between vApp Networks

For an example of a static routing service in a vApp network, see Network Services in vApp Networks.

Load Balancer Service Configurations

An Edge Gateway can provide load-balancing services that allow you to distribute incoming requests to a specific external IP address across multiple internal IP addresses. Several load-balancing algorithms are supported.

A load balancer service provides load balancing for TCP, HTTP, and HTTPS traffic. The load balancer accepts incoming IP requests on an external or internal interface, and uses the algorithm you specify to distribute requests across a pool of servers.

To add a load-balancer service to an Edge Gateway, include a `LoadBalancerService` element in the Edge Gateway's `EdgeGatewayServiceConfiguration`.

Example: Load Balancer Service

This fragment of an `EdgeGatewayServiceConfiguration` defines a `LoadBalancerService` that accepts incoming requests at external address `https://192.168.1.100` and balances them across two servers at internal addresses `10.200.100.10` and `10.200.100.11`. The following elements define a `LoadBalancerService`:

- A `Pool` that contains `ServicePort` and `Member` elements. A `LoadBalancerService` must include a `Pool` that defines a `ServicePort` for each protocol on which the load balancer handles incoming requests. You can define up to three `ServicePort` elements, one for each supported protocol (HTTP, HTTPS, TCP). This load balancer handles only HTTPS requests, so it requires only one `ServicePort` element in its `Pool`.

  You must specify one of the following load-balancing algorithms in the `Algorithm` element of the `ServicePort`.

  IP_HASH

  Selects a server based on a hash of the source and destination IP address of each packet.

  LEAST_CONN

  Distributes client requests to multiple servers based on the number of connections already on the server. New connections are sent to the server with the fewest connections.

ROUND_ROBIN

Each server is used in turn according to the weight assigned to it. This is the smoothest and fairest algorithm when the server's processing time remains equally distributed.

URI

The request URL is hashed and divided by the total weight of the running servers. (If the request URL includes a query, it is discarded, and only the fragment of the URL to the left of the ? is considered.) The result designates which server receives the request, ensuring that a request is always directed to the same server as long as all servers remain available.

The `Pool` in this example also defines an optional `HealthCheck` element that specifies parameters used for periodic verification that all pool members are responding to requests.

Each `Member` element in the `Pool` specifies the `IpAddress` of a virtual machine that provides the service being requested. Incoming requests are balanced across all members of the pool. Because the `Algorithm` specified for this Pool is `ROUND_ROBIN`, each `Member` must be assigned a `Weight`.

- A `VirtualServer` element that defines the `Interface`, an internal or external interface defined by the containing `EdgeGateway`, on which requests are accepted. The network referenced in the `Interface` element must be configured with an IP sub-allocation.

Note Each `Member` of a load balancer `Pool` can contain its own `ServicePort` element. If this element is present in a `Member`, its contents override the `ServicePort` element of the Pool.

For more information about `LoadBalancerService` elements and attributes, see the schema reference.

```
<LoadBalancerService>
   <IsEnabled>true</IsEnabled>
   <Pool>
      <Name>HTTPS_pool</Name>
      <ServicePort>
         <IsEnabled>true</IsEnabled>
         <Protocol>HTTPS</Protocol>
         <Algorithm>ROUND_ROBIN</Algorithm>
         <Port>443</Port>
         <HealthCheck>
            <Mode>TCP</Mode>
            <HealthThreshold>2</HealthThreshold>
            <UnhealthThreshold>3</UnhealthThreshold>
            <Interval>5</Interval>
            <Timeout>15</Timeout>
         </HealthCheck>
      </ServicePort>
      <Member>
         <IpAddress>10.200.100.10</IpAddress>
         <Weight>1</Weight>
      </Member>
      <Member>
```

```
         <IpAddress>10.200.100.11</IpAddress>
         <Weight>1</Weight>
      </Member>
   </Pool>
   <VirtualServer>
      <IsEnabled>true</IsEnabled>
      <Name>Example Virtual Server</Name>
      <Description>Incoming LoadBalancerService Requests</Description>
      <Interface
         type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
         href="https://vcloud.example.com/api/admin/network/199" />
      <IpAddress>192.168.1.100</IpAddress>
      <ServiceProfile>
         <IsEnabled>true</IsEnabled>
         <Protocol>HTTPS</Protocol>
         <Port>443</Port>
         <Persistence>
            <Method>SSL_SESSION_ID</Method>
         </Persistence>
      </ServiceProfile>
      <Logging>true</Logging>
      <Pool>HTTPS_pool</Pool>
   </VirtualServer>
</LoadBalancerService>
```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See Configure Services on an Edge Gateway.

IPsec VPN Service Configurations

An Edge Gateway configuration can define an IPsec virtual private networking (VPN) service to provide secure virtual private networking within an organization, between organization VDC networks, or between an organization VDC network and an external IP address.

An `EdgeGateway` can contain zero or more `GatewayIpsecVpnService` elements, each of which defines VPN tunnels and endpoints.

Example: IPsec VPN Service in an Edge Gateway

```
<GatewayIpsecVpnService>
   <IsEnabled>true</IsEnabled>
   <Tunnel>
      <Name>Example VPN Tunnel</Name>
      <Description />
      <IpsecVpnLocalPeer>
         <Id>3786bb05-dc9a-471b-91cd-554499d45629</Id>
         <Name>gw02</Name>
      </IpsecVpnLocalPeer>
      <PeerIpAddress>10.147.46.68</PeerIpAddress>
      <PeerId>C64E127E-5E86-C57C-17ED-EB175A7A1811</PeerId>
      <LocalIpAddress>10.147.46.66</LocalIpAddress>
      <LocalId>6844BBB4-24E6-7A50-0F29-EB175A7AD899</LocalId>
      <LocalSubnet>
         <Name>nw01</Name>
         <Gateway>192.168.1.1</Gateway>
         <Netmask>255.255.255.0</Netmask>
      </LocalSubnet>
      <PeerSubnet>
         <Name>nw02</Name>
         <Gateway>192.168.2.1</Gateway>
         <Netmask>255.255.255.0</Netmask>
      </PeerSubnet>
      <SharedSecret>L3hithJa3zH7K4q2tH...</SharedSecret>
      <SharedSecretEncrypted>false</SharedSecretEncrypted>
      <EncryptionProtocol>AES256</EncryptionProtocol>
      <Mtu>1500</Mtu>
      <IsEnabled>true</IsEnabled>
   </Tunnel>
</GatewayIpsecVpnService>
```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See Configure Services on an Edge Gateway.

Gateway DHCP Service Configurations

An Edge Gateway configuration can define a DHCP service that assigns IP addresses to clients on organization VDC networks. You can configure multiple address pools, each of which defines a range of IP addresses that are reserved for a specific network.

DHCP services for organization VDC networks are provided by the Edge Gateway to which those network connect. An `EdgeGateway` can contain an more `GatewayDhcpService` element. The service can define a pool of IP addresses for each organization VDC connected to the Edge Gateway.

Note To provide DHCP services on a vApp network, use the `DhcpService` element.

Example: Gateway DHCP Service

The following fragment of an `EdgeGatewayServiceConfiguration` defines and enables a `GatewayDhcpService` that defines two Pool objects, each of which allocates a range of IP addresses for a single organization VDC network.

```
<GatewayDhcpService>
   <IsEnabled>true</IsEnabled>
   <Pool>
      <IsEnabled>true</IsEnabled>
      <Network
         type="application/vnd.vmware.admin.orgVdcNetwork+xml"
         href="https://vcloud.example.com/api/admin/network/46" />
      <DefaultLeaseTime>3600</DefaultLeaseTime>
      <MaxLeaseTime>7200</MaxLeaseTime>
      <LowIpAddress>10.100.1.100</LowIpAddress>
      <HighIpAddress>10.100.1.150</HighIpAddress>
   </Pool>
   <Pool>
      <IsEnabled>true</IsEnabled>
      <Network
         type="application/vnd.vmware.admin.orgVdcNetwork+xml"
         href="https://vcloud.example.com/api/admin/network/47" />
      <DefaultLeaseTime>3600</DefaultLeaseTime>
      <MaxLeaseTime>7200</MaxLeaseTime>
      <LowIpAddress>10.100.1.200</LowIpAddress>
      <HighIpAddress>10.100.1.250</HighIpAddress>
   </Pool>
</GatewayDhcpService>
```

To add this service to an Edge Gateway, include it in an `EdgeGatewayServiceConfiguration`. See Configure Services on an Edge Gateway.

Create an Organization VDC Network

To add a network to an organization VDC, an administrator POSTs an `OrgVdcNetwork` element to the VDC's `add` URL for `networks`. An organization VDC network with a direct connection to an external network must be created by a system administrator. All other organization VDC network types can be created by either a system administrator or an organization administrator.

The contents of the `Configuration` element of the `OrgVdcNetwork` define the properties of the network, including its connections to other networks.

For more information about the types of networks you can create and the resources on which they depend, see About VMware Cloud Director Networks.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the organization VDC to which you want to add the network.

   This request retrieves the admin view of an organization VDC.

   ```
   GET https://vcloud.example.com/api/admin/vdc/44
   ```

2 Examine the response to locate the `Link` element that contains the URL for adding networks to the VDC.

   This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.orgVdcNetwork+xml`, as shown here:

   ```
   <Link
      rel="add"
      type="application/vnd.vmware.admin.orgVdcNetwork+xml"
      href="https://vcloud.example.com/api/admin/vdc/44/networks" />
   ```

3 Create an `OrgVdcNetwork` element.

4 POST the `OrgVdcNetwork` element to the URL described in Step 2.

Results

The server takes the requested action and returns a `Task` element that tracks the progress of the request. The `Owner` element of this task includes the `href` attribute of the new network. When the task completes, you can use the value of this attribute with a GET request to retrieve the XML representation of the new network.

Create an Organization VDC Network With a Routed Connection

An organization VDC network with a routed connection provides controlled access to machines and networks outside of the organization VDC. System administrators and organization administrators can configure network address translation (NAT) and firewall settings on the network's Edge Gateway to make specific virtual machines in the VDC accessible from an external network.

You can create an IPv4 or IPv6 routed network.

Prerequisites

- This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Retrieve the list of Edge Gateways in the organization VDC in which you plan to create the routed network.

  You can use a query like this one, where *href* is the value of the `href` attribute of your organization VDC:

  ```
  https://vcloud.example.com/api/query?type=edgeGateway&format=records&filter=vdc==href
  ```

  If this organization VDC does not contain any Edge Gateways, or does not contain an Edge Gateway that has the configuration you want, a system administrator can create a new Edge Gateway. See Create an Edge Gateway.

2 Choose an Edge Gateway that has interface capacity available.

  An Edge Gateway can support a maximum of nine internal and external interfaces. At least one of those interfaces is typically consumed by a connection to an external network. Creation of a routed organization VDC network requires the Edge Gateway to have an unused interface available for the new network. To see how many interfaces each Edge Gateway in your organization VDC is using, you can run the query shown in Step 1, then add the values of the `numberOfExtNetworks` and `numberOfOrgNetworks` attributes. If the total is less than 9, the Edge Gateway can accommodate a new routed organization VDC network.

3 Create an `OrgVdcNetwork` element.

   a Specify a value of `natRouted` in the `FenceMode` element of the network `Configuration`.

      You can specify additional `Configuration` parameters, as noted in the schema reference.

   b Specify the `href` of the Edge Gateway you chose in Step 2 in the `EdgeGateway` element.

   See the request portion of Create an Organization VDC Network With a Routed Connection.

4 POST the `OrgVdcNetwork` element to the URL for adding networks to the organization VDC.

   See the request portion of Create an Organization VDC Network With a Routed Connection

Results

The server takes the requested action and returns an XML representation of the partially-created object. This representation includes an `href` attribute, properties specified in the creation request, and an embedded `Task` element that tracks the creation of the object. When the task completes, the object has been created, and you can use the value of the `href` attribute with a GET request to retrieve the XML representation of the object.

See the response portion of Create an Organization VDC Network With a Routed Connection.

Example: Create an Organization VDC Network With a Routed Connection

This example adds a routed network to the organization VDC created in Add a VDC to an Organization . The network connects through the Edge Gateway created in Create an Edge Gateway. Because the creation request sets the value of the `IsShared` element to `true`, the new network is made available in all VDCs in this organization.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/networks
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgVdcNetwork
   name="RoutedOVDCNet"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Routed through an Edge Gateway</Description>
   <Configuration>
      <IpScopes>
         <IpScope>
            <IsInherited>false</IsInherited>
            <Gateway>192.168.0.1</Gateway>
            <Netmask>255.255.255.0</Netmask>
            <Dns1>10.147.115.1</Dns1>
            <DnsSuffix>example.com</DnsSuffix>
            <IpRanges>
               <IpRange>
                  <StartAddress>192.168.0.100</StartAddress>
                  <EndAddress>192.168.0.199</EndAddress>
               </IpRange>
            </IpRanges>
         </IpScope>
      </IpScopes>
      <FenceMode>natRouted</FenceMode>
   </Configuration>
   <EdgeGateway
      href="https://vcloud.example.com/api/admin/gateway/2000" />
   <IsShared>true</IsShared>
</OrgVdcNetwork>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<OrgVdcNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="RoutedOVDCNet"
   type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
   href="https://vcloud.example.com/api/admin/network/59" ...>
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
      href="https://vcloud.example.com/api/admin/network/59" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/network/59" />
   <Link
      rel="up"
      type="application/vnd.vmware.admin.vdc+xml"
      href="https://vcloud.example.com/api/admin/vdc/44" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/network/59/metadata" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.allocatedNetworkAddress+xml"
      href="https://vcloud.example.com/api/admin/network/59/allocatedAddresses/" />
   <Description>Routed through an Edge Gateway</Description>
   <Tasks>
      <Task
         status="running"
         ...
         operation="Creating Network RoutedOVDCNet(59)"
         ...
         href="https://vcloud.example.com/api/task/999">
      </Task>
   </Tasks>
   <Configuration>
      ...
      <RetainNetInfoAcrossDeployments>false</RetainNetInfoAcrossDeployments>
   </Configuration>
   <EdgeGateway
      type="application/vnd.vmware.admin.edgeGateway+xml"
      name="theEdge"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000" />
   <IsShared>true</IsShared>
</OrgVdcNetwork>
```

Note When the `Task` completes, the new network is represented in the `EdgeGateway` by a `GatewayInterface` whose `InterfaceType` is `Internal`. Unlike the `Uplink` interface that you create when you create an `EdgeGateway`, an internal interface cannot be created explicitly. It is created only as a side-effect of creating a routed organization VDC network.

Create an Isolated Organization VDC Network

An isolated organization VDC network provides an isolated, private network that machines in the organization VDC can connect to. This network provides no connectivity to machines outside this organization VDC.

You can create only an IPv4 isolated organization VDC network.

Prerequisites

- This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Create an `OrgNetwork` element.

   Specify a value of `isolated` in the `FenceMode` element of the network `Configuration`. See the request portion of Create an Isolated Organization VDC Network.

2 POST the `OrgNetwork` element you created in Step 1 to the URL for adding networks to the organization VDC

   See the request portion of Create an Isolated Organization VDC Network.

Results

The server takes the requested action and returns an XML representation of the partially-created object. This representation includes an `href` attribute, properties specified in the creation request, and an embedded `Task` element that tracks the creation of the object. When the task completes, the object has been created, and you can use the value of the `href` attribute with a GET request to retrieve the XML representation of the object.

See the response portion of Create an Isolated Organization VDC Network.

Example: Create an Isolated Organization VDC Network

This example adds an isolated network to the organization VDC created in Add a VDC to an Organization . It includes a `ServiceConfig` element that configures a DHCP service for the network. This type of DHCP service is identical to the DHCP service supported for a vApp network, and can specify only a single IP address range. No other network services can be created in an isolated organization VDC network.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/networks
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgVdcNetwork
   name="Isolated"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Isolated Organization VDC Network</Description>
   <Configuration>
      <IpScopes>
         <IpScope>
            <IsInherited>false</IsInherited>
            <Gateway>192.168.0.1</Gateway>
            <Netmask>255.255.255.0</Netmask>
            <Dns1>10.147.115.1</Dns1>
            <DnsSuffix>example.com</DnsSuffix>
            <IpRanges>
               <IpRange>
                  <StartAddress>192.168.0.100</StartAddress>
                  <EndAddress>192.168.0.199</EndAddress>
               </IpRange>
            </IpRanges>
         </IpScope>
      </IpScopes>
      <FenceMode>isolated</FenceMode>
   </Configuration>
   <ServiceConfig>
      <DhcpService>
         <IsEnabled>false</IsEnabled>
         <DefaultLeaseTime>3600</DefaultLeaseTime>
         <MaxLeaseTime>7200</MaxLeaseTime>
         <IpRange>
            <StartAddress>192.168.0.2</StartAddress>
            <EndAddress>192.168.0.99</EndAddress>
         </IpRange>
      </DhcpService>
   </ServiceConfig>
</OrgVdcNetwork>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<OrgVdcNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Isolated"
   type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
   href="https://vcloud.example.com/api/admin/network/60"...>
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
      href="https://vcloud.example.com/api/admin/network/60" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/network/60" />
   <Link
      rel="up"
      type="application/vnd.vmware.admin.vdc+xml"
      href="https://vcloud.example.com/api/admin/vdc/44" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/network/60/metadata" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.allocatedNetworkAddress+xml"
      href="https://vcloud.example.com/api/admin/network/60/allocatedAddresses/" />
   <Description>Isolated Organization Vdc Network</Description>
   <Tasks>
      <Task
         name="task"
         status="running"
         operation="Creating Network Isolated(60)"
         ...
      </Task>
   </Tasks>
   <Configuration>
      ...
   </Configuration>
   <ServiceConfig>
      <DhcpService>
         <IsEnabled>false</IsEnabled>
         <DefaultLeaseTime>3600</DefaultLeaseTime>
         <MaxLeaseTime>7200</MaxLeaseTime>
         <IpRange>
            <StartAddress>192.168.0.2</StartAddress>
            <EndAddress>192.168.0.99</EndAddress>
         </IpRange>
      </DhcpService>
   </ServiceConfig>
</OrgVdcNetwork>
```

Synchronize Syslog Server Settings for an Edge Gateway or vApp Network

When you change the IP addresses of the primary or secondary `syslog` server for a cloud, you must also synchronize the `syslog` server settings for each Edge Gateway or vApp network that includes a `FirewallService` for which logging is enabled.

If a system administrator changes the `SyslogServerSettings` for a cloud, all Edge Gateways and vApp networks that are configured with a firewall service whose `EnableLogging` element has a value of `true` must be synchronized with the new syslog server settings so that logging to the system syslog server can continue without interruption.

Note You do not need to synchronize syslog server settings for an Edge Gateway when you change its `TenantSyslogServerSettings`.

Prerequisites

- To synchronize syslog server settings for an Edge Gateway, you must be an organization administrator or system administrator.
- To synchronize syslog server settings for a vApp network, you must be the vApp owner.

Procedure

1 Retrieve the XML representation of the vApp network or Edge Gateway.

2 Examine the response to locate the `Link` element that contains the URL for the `syncSyslogServerSettings` action.

  This element has a `rel` attribute value of `syncSyslogSettings` and a `type` attribute value of `application/vnd.vmware.vcloud.task+xml`, as shown in this excerpt:

```
<Link
   rel="syncSyslogSettings"
   type="application/vnd.vmware.vcloud.task+xml"
   href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/
syncSyslogServerSettings"
```

3 Make a POST request to the `href` value of the link described in Step 2.

  The request does not have a request body. The response is a task.

Example: Synchronize Syslog Server Settings for an Edge Gateway

This request synchronizes the syslog server settings for the Edge Gateway created in Create an Edge Gateway.

```
POST https://vcloud.example.com/api/admin/edgeGateway/2000/action/syncSyslogServerSettings
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ...
   status="running"
   ...
   operation="Synchronized syslog settings for EdgeGateway theEdge(2000)>
   ...>
</Task>
```

Catalog Administration

A newly created organization has no catalogs in it. After an organization administrator or catalog author creates a catalog, members of the organization can use it as a destination for uploads or a source of subscription-based content.

Catalogs contain references to vApp templates and media images. You can configure a catalog in several different ways:

- as a repository for local content that can remain private to the catalog owner or can be shared with other users, groups, or organizations in your cloud
- as a source of published content, to which other clouds can subscribe.
- as a local repository for content published by another cloud or any Web site that hosts a VMware Content Subscription Protocol (VCSP) endpoint.

An organization administrator or catalog owner controls catalog sharing. Organization administrators in organizations that have permission to publish catalogs control publication and subscription options for catalogs in their organization. A system administrator can enable background synchronization of catalogs with external sources and set background synchronization schedules to regulate consumption of network bandwidth by this activity.

Access to Catalogs

A catalog initially grants full control to its owner and no access to other users. The catalog owner or a user with organization administrator or catalog author rights can grant catalog access to other members of the organization, individually or collectively. See Controlling Access to vApps and Catalogs. Organization administrators and system administrators can share a catalog with other organizations in the cloud.

Synchronization

The VMware Content Subscription Protocol (VCSP) is an open standard that can be implemented by any system that can provide HTTP or HTTPS access. Because VMware Cloud Director implements this protocol, catalogs can subscribe to content that originates in another instance of VMware Cloud Director or at any remote site that supports a VCSP endpoint. When content at a remote site changes, you must synchronize the catalog items that hold local copies of that content. Synchronization keeps a catalog up to date with its external subscription. A catalog owner can synchronize individual catalog items or entire catalogs at any time. A system administrator can also schedule background synchronization for catalogs, so that all externally subscribed catalogs in the system are synchronized on a common schedule.

Version Numbers

As part of VCSP support, catalogs and catalog items have version numbers, which are integer values that increment monotonically. The catalog item version number increases whenever any of the following changes occur.

- A file in the referenced entity is added, removed, or changed.
- The name or description of the catalog item is changed.

The catalog version number increases whenever any of the following changes occur.

- A catalog item is added to or removed from the catalog.
- The version number of any contained catalog item changes.
- The name or description of the catalog is changed.

A catalog with an external subscription contains the most recent version of each of its catalog items if it has been synchronized with its external source.

Add a Catalog to an Organization

Every organization has an `add` URL for `catalogs`. An administrator or catalog author can create a catalog by POSTing an `AdminCatalog` element to this URL

A new `Catalog` object is an empty container for catalog items, which are references to vApp templates and media images. There are several options for creating a catalog:

- You can create a catalog to use in your own organization or cloud, to hold content that you create locally by uploading, importing, or capturing.
- If the system administrator has given your organization permission to publish externally, you can create a catalog that is published externally or enable external publication of a catalog that you create locally. See Create a Catalog For External Publication and Publish an Existing Catalog Externally.
- If the system administrator has given your organization permission to subscribe to an external catalog, you can create a catalog that downloads its contents from an external source.

  Note You cannot use catalogs created from an external subscription to hold catalog items that you create locally. Catalogs that contain catalog items created locally cannot have an external subscription. See Create a Catalog With an External Subscription.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that at least one VDC exists in your organization. You cannot create a catalog in an organization that has no VDCs.

Procedure

1 Retrieve the XML representation of the organization to which to add the catalog.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26
```

2 Examine the response to locate the `Link` element that contains the URL for adding catalogs to the organization.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.catalog+xml`, as shown here:

```
<Link
   href="https://vcloud.example.com/api/admin/org/5/catalogs"
   rel="add"
   type="application/vnd.vmware.admin.catalog+xml"/>
```

3 Create an `AdminCatalog` element.

See the request portion of Create a Catalog.

4 POST the `AdminCatalog` element to the organization's `add` URL for `catalogs`.

See the request portion of Create a Catalog.

Results

The server creates an empty catalog and returns its representation in the response. See the response portion of Create a Catalog.

Example: Create a Catalog

This example adds a catalog to the organization created in Create an Organization. Because the request does not specify a `CatalogStorageProfile`, the catalog is created on the default storage profile for the first VDC created in the organization. To create the catalog on a specific storage profile, you can add a `CatalogStorageProfiles` element to the request. See Specify a Storage Profile for a Catalog.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/catalogs
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminCatalog
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Example Catalog">
   <Description>New Catalog for Example Org</Description>
</AdminCatalog>
```

The response contains information extracted from the request, and includes these additions that the server creates:

- A URL, in the value of the `href` attribute of the response body, that references the new catalog.
- Links that implement operations on the catalog.
- A link to an alternate view of this catalog. All users can access the catalog at this URL.
- An empty `CatalogItems` element.
- A `Task` that tracks the creation of the catalog.
- An `IsPublished` element whose content is the string `false`, indicating that the catalog is not shared with other organizations. See Share a Catalog with All Organizations in a Cloud.
- A `VersionNumber` element with an initial value of `1`. See Version Numbers.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<AdminCatalog
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="Example Catalog"
   id="urn:vcloud:catalog:32"
   type="application/vnd.vmware.admin.catalog+xml"
   href="https://vcloud.example.com/api/admin/catalog/32">
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.catalog+xml"
      href="https://vcloud.example.com/api/catalog/32" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.owner+xml"
      href="https://vcloud.example.com/api/admin/catalog/32/owner" />
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.catalog+xml"
      href="https://vcloud.example.com/api/admin/catalog/32" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/catalog/32" />
   <Link ... >
   ...
   <Tasks>
      <Task
         status="running"
         ...
         operation="Creating Catalog Example Catalog (32)"
```

```
          ...>
          ...
       <Task>
    <Tasks>
    <Description>New Catalog for Example Org</Description>
    <CatalogItems/>
    <IsPublished>false</IsPublished>
    <DateCreated>2013-06-18T09:52:59.260-07:00</DateCreated>
    <VersionNumber>1</VersionNumber>
    <CatalogStorageProfiles/>
</AdminCatalog>
```

Create a Catalog For External Publication

An organization administrator in an organization that has permission to publish externally can create a catalog whose contents, catalog items created locally, are made available to external consumers on a subscription basis.

Organizations that are permitted to publish externally can enable any of their catalogs for external publication. A catalog that is enabled for external publication becomes accessible at a URL assigned by the system, and can be protected by a password. A catalog in another instance of VMware Cloud Director can subscribe to an externally published catalog and maintain an up-to-date set of catalog items. See Synchronization.

You can use this procedure to create a new catalog that is enabled for external publication, or you can use the procedure shown in Publish an Existing Catalog Externally to enable an existing catalog for external publication. If a catalog has an external subscription, you cannot enable it for external publication.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that at least one VDC exists in your organization. You cannot create a catalog in an organization that has no VDCs.
- Verify that your organization has permission to publish externally.

  The `OrgGeneralSettings` in the `AdminOrg` element that represents your organization must have a `CanPublishExternally` element with a value of `true`.

  ```
  <CanPublishExternally>true</CanPublishExternally>
  ```

Retrieve the XML representation of the organization to which to add the catalog, and examine the response to locate the `Link` element that contains the URL for adding catalogs to the organization. See Add a Catalog to an Organization .

Procedure

1 Create an AdminCatalog element that includes PublishExternalCatalogParams.

  See the request portion of Create a Catalog For External Publication.

2 POST the AdminCatalog element to the organization's add URL for catalogs.

  See the request portion of Create a Catalog For External Publication.

Results

The server creates an empty catalog and returns its representation in the response. See the response portion of Create a Catalog.

Example: Create a Catalog For External Publication

This request is similar to the one used in Create a Catalog but includes a PublishExternalCatalogParams element that specifies that the catalog is to be published externally. This request creates the catalog with a content cache, which can improve synchronization performance for external catalogs that subscribe to this one, but requires enough space on the system's transfer storage to accommodate all of the items in the catalog. This request also sets the PreserveIdentityInfoFlag element to false, which prevents values such as the BIOS UUID and MAC address associated with virtual machine identities from being published. If you omit this element or set its value to true, these values are published for all catalog items that represent virtual machines.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/catalogs
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminCatalog
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="PublishedExternally">
   <Description>Example Catalog for External Publication</Description>
   <PublishExternalCatalogParams>
      <IsPublishedExternally>true</IsPublishedExternally>
      <IsCacheEnabled>true</IsCacheEnabled>
      <PreserveIdentityInfoFlag>false</PreserveIdentityInfoFlag>
      <Password>Pa55w0rd</Password>
   </PublishExternalCatalogParams>
</AdminCatalog>
```

The response is similar to the one shown in Create a Catalog, but includes the PublishExternalCatalogParams element. The embedded Task element tracks the creation of the catalog and its accompanying Web site. The Password element in the request is never returned.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<AdminCatalog
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="PublishedExternally"
   id="urn:vcloud:catalog:33"
   type="application/vnd.vmware.admin.catalog+xml"
   href="https://vcloud.example.com/api/admin/catalog/33">
   ...
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link ... >
   ...
   <Tasks>
      <Task
         status="running"
         ...
         operation="Creating Catalog PublishedExternally (33)"
         ...>
         ...
      <Task>
   <Tasks>
   <Description>Example Catalog for External Publication</Description>
   <CatalogItems/>
   <IsPublished>false</IsPublished>
   <DateCreated>2013-06-18T09:55:59.131-07:00</DateCreated>
   <VersionNumber>1</VersionNumber>
   <CatalogStorageProfiles/>
   <PublishExternalCatalogParams>
      <IsPublishedExternally>true</IsPublishedExternally>
      <IsCacheEnabled>true</IsCacheEnabled>
      <PreserveIdentityInfoFlag>false</PreserveIdentityInfoFlag>
   </PublishExternalCatalogParams>
</AdminCatalog>
```

When the task completes, the `PublishExternalCatalogParams` includes a URL at which external consumers can connect to the catalog.

```
<AdminCatalog ... >
   ...
   <PublishExternalCatalogParams>
      <IsPublishedExternally>true</IsPublishedExternally>
      <IsCacheEnabled>true</IsCacheEnabled>
    <CatalogPublishedUrl>https://vcloud2.example.com/vcsp/catalog/5</CatalogPublishedUrl>
   </PublishExternalCatalogParams>
</AdminCatalog>
```

Create a Catalog With an External Subscription

An organization administrator in an organization that has permission to subscribe can create a catalog whose contents are downloaded from an external source. You cannot use catalogs created this way to hold catalog items that are created locally.

Organizations that are permitted to subscribe to external sources can create catalogs whose contents are downloaded from a catalog hosted on another instance of VMware Cloud Director, or any Web site that implements the VMware Content Subscription Protocol (VCSP) . See Synchronization.

If a catalog has an external subscription, you cannot enable it for external publication.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that at least one VDC exists in your organization. You cannot create a catalog in an organization that has no VDCs.
- Verify that your organization has permission to create catalogs with external subscriptions.

  The `OrgGeneralSettings` in the `AdminOrg` element that represents your organization must have a `CanSubscribe` element with a value of `true`.

  ```
  <CanSubscribe>true</CanSubscribe>
  ```

- The external source must implement the VMware Content Subscription Protocol. See Synchronization.
- You must know the URL of the external source, and the password if one is required.

Retrieve the XML representation of the organization to which you want to add the catalog, and examine the response to locate the `Link` element that contains the URL for adding catalogs to the organization. See Add a Catalog to an Organization .

Procedure

1. Create an `AdminCatalog` element that includes `ExternalCatalogSubscriptionParams`.

   See the request portion of Create a Catalog With an External Subscription.

2. POST the `AdminCatalog` element to the organization's `add` URL for `catalogs`.

   See the request portion of Create a Catalog With an External Subscription.

Results

The server creates an empty catalog and returns its representation in the response. See the response portion of Create a Catalog.

Example: Create a Catalog With an External Subscription

This request creates a catalog with a subscription to the VCSP URL https://vcloud2.example.com/vcsp/catalog/5. Because the `LocalCopy` element in `ExternalCatalogSubscriptionParams` has a value of `false`, files that comprise a vApp template or media image that a catalog item references are not downloaded until a user requests them. If you create a catalog where `LocalCopy` has a value of `true`, these files are downloaded the first time that the catalog subscription is synchronized, and on each subsequent synchronization where any of the catalog items has a newer version number. The default value of `LocalCopy` is`false`.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/catalogs
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminCatalog
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="SubscribedExternally">
   <Description>Example Catalog With External Subscription</Description>
   <ExternalCatalogSubscriptionParams>
      <SubscribeToExternalFeeds>true</SubscribeToExternalFeeds>
      <Location>https://vcloud2.example.com/vcsp/catalog/5</Location>
      <Password>Pa55w0rd</Password>
      <LocalCopy>false</LocalCopy>
   </ExternalCatalogSubscriptionParams>
</AdminCatalog>
```

The response is similar to the one shown in Create a Catalog, but includes the `ExternalCatalogSubscriptionParams` element that you supplied in the request. The `Password` element in the request is never returned.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<AdminCatalog
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="SubscribedExternally"
   id="urn:vcloud:catalog:34"
   type="application/vnd.vmware.admin.catalog+xml"
   href="https://vcloud.example.com/api/admin/catalog/34">
   ...
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link ... >
   <Link
      rel="sync"
      href="https://vcloud.example.com/api/catalog/34/action/sync"/>
   <Link ... >
   ...
   <Tasks>
      <Task
         status="running"
         ...
         operation="Creating Catalog SubscribedExternally (34)"
         ...
      <Task>
   <Tasks>
   <Description>Example Catalog With External Subscription</Description>
   <CatalogItems/>
   <IsPublished>false</IsPublished>
   <DateCreated>2013-06-18T10:00:03.012-07:00</DateCreated>
   <VersionNumber>1</VersionNumber>
   <CatalogStorageProfiles/>
   <ExternalCatalogSubscriptionParams>
      <SubscribeToExternalFeeds>true</SubscribeToExternalFeeds>
      <Location>https://vcloud2.example.com/vcsp/catalog/5</Location>
      <LocalCopy>false</LocalCopy>
   </ExternalCatalogSubscriptionParams>
</AdminCatalog>
```

What to do next

Synchronize the catalog with its external source. Any catalog with an external subscription includes a link of the form:

```
<Link
   rel="sync"
   href="https://vcloud.example.com/api/catalog/id/action/sync"/>
```

As a catalog owner or administrator, you can POST a request to the `rel="sync"` URL to synchronize the catalog with its external source. See Synchronize a Catalog or Catalog Item.

Specify a Storage Profile for a Catalog

Any request that creates a catalog can also specify a storage profile for the items in that catalog.

If you do not specify a `CatalogStorageProfile` when creating a catalog, the catalog is created on the default storage profile for the first VDC created in the organization. To create the catalog on a specific storage profile, retrieve the list of available storage profiles in your organization and specify one in the `CatalogStorageProfiles` element of the request.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that at least one VDC exists in your organization. You cannot create a catalog in an organization that has no VDCs.

Procedure

1 Find the list of storage profiles in your organization.

   Use a query like this one.

   ```
   https://vcloud.example.com/api/query?type=orgVdcStorageProfile&format=references
   ```

   The response is a list of references to all the storage profiles used by VDCs in your organization.

   ```
   <OrgVdcStorageProfileReferences ... >
      <OrgVdcStorageProfileReference
         type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
         name="*"
         id="urn:vcloud:vdcstorageProfile:59"
         href="https://vcloud.example.com/api/vdcStorageProfile/59" />
      <OrgVdcStorageProfileReference
         type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
         name="Silver"
         id="urn:vcloud:vdcstorageProfile:84"
         href="https://vcloud.example.com/api/vdcStorageProfile/84" />
   </OrgVdcStorageProfileReferences>
   ```

   Note If you are a system administrator, you must filter the response to show only those storage profiles in the organization where you are creating the catalog.

2 Create an `AdminCatalog` element that includes a `CatalogStorageProfiles` element.

   You can include a single `CatalogStorageProfile` in the request. Only the `type` and `href` attributes are required. See the request portion of Specify a Storage Profile for a Catalog.

3 POST the `AdminCatalog` element to the organization's `add` URL for `catalogs`.

   See the request portion of Specify a Storage Profile for a Catalog.

Results

The server creates an empty catalog on the specified storage profile and returns its representation in the response. See the response portion of Create a Catalog.

Example: Specify a Storage Profile for a Catalog

This example modifies the request shown in Create a Catalog to add a `CatalogStorageProfiles` element..

Request:

```
POST https://vcloud.example.com/api/admin/org/26/catalogs
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminCatalog
   xmlns="http://www.vmware.com/vcloud/v1.5"
```

```
   name="Example Catalog">
   <Description>New Catalog for Example Org</Description>
   <CatalogStorageProfiles>
      <VdcStorageProfile
         type="application/vnd.vmware.admin.vdcStorageProfile+xml"
         href="https://vcloud.example.com/api/vdcStorageProfile/59" />
   </CatalogStorageProfiles>
</AdminCatalog>
```

The response is similar to the one shown in Create a Catalog, and includes a `CatalogStorageProfiles` element derived form the one that you specified in the request.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.catalog+xml
...
<AdminCatalog
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="Example Catalog"
   ...

   <Description>New Catalog for Example Org</Description>
   <CatalogItems/>
   ...
   <CatalogStorageProfiles>
      <VdcStorageProfile
         type="application/vnd.vmware.admin.vdcStorageProfile+xml"
         name="*"
         id="urn:vcloud:vdcstorageProfile:59"
         href="https://vcloud.example.com/api/vdcStorageProfile/59" />
   </CatalogStorageProfiles>
</AdminCatalog>
```

Update Catalog Access Controls

If you are an administrator or catalog owner, you can use a catalog's `controlAccess` links to grant or restrict access to the catalog.

A catalog initially grants full access to its owner and no access to other users. An administrator or the catalog owner can use the VMware Cloud Director API access control mechanism to view or modify catalog access controls. For a general discussion of access controls in VMware Cloud Director, see Controlling Access to vApps and Catalogs.

Procedure

1 Retrieve the XML representation of the catalog.

Use a request like this one:

```
GET https://vcloud.example.com/api/catalog/id
```

2 Examine the `Catalog` response to find the `controlAccess` links that it contains.

These links have the following form:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.controlAccess+xml"
   href="https://vcloud.example.com/api/catalog/id/controlAccess/"/>
<Link
   rel="controlAccess"
   type="application/vnd.vmware.vcloud.controlAccess+xml"
   href="https://vcloud.example.com/api/catalog/id/action/controlAccess/"/>
```

3 Create a `ControlAccessParams` element request body that specifies the details of the update.

4 POST the `ControlAccessParams` element to the `action/controlAccess` link for the catalog.

Example: Update Catalog Access Controls

This request updates the access controls of a catalog to grant full control to one user and read-only access to another user. The request body, a `ControlAccessParams` element, specifies a value of `false` for the `IsSharedToEveryone` element, and contains an `AccessSetting` element for each user whose access rights are being modified. Each user is identified by a reference to a `User` object. See User and Group Administration. The response, a subset of which appears in this example, echoes the request.

Request:

```
POST https://vcloud.example.com/api/org/9/catalog/32/action/controlAccess
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/40"/>
         <AccessLevel>FullControl</AccessLevel>
      </AccessSetting>
      <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.user+xml"
            href="https://vcloud.example.com/api/admin/user/45"/>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
   </AccessSettings>
</ControlAccessParams>
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      ...
   </AccessSettings>
</ControlAccessParams>
```

Share a Catalog with All Organizations in a Cloud

An organization administrator can share a catalog to make it visible to the administrators in all other organizations in a cloud.

The owner of a catalog can make it available to other users in the containing organization. See Update Catalog Access Controls. If you are an organization administrator, you can also share catalogs with all other organizations in your cloud if your organization's `CanPublishCatalogs` element has a value of `true`. The value of this element is controlled by the system administrator. To share a catalog, make a POST request to the catalog's `action/publish` URL and supply a `PublishCatalogParams` body that sets the value of the catalog's `IsPublished` element to true.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the catalog to share.

   Use a request like this one, where *id* is the identifier of the catalog:

   ```
   GET https://vcloud.example.com/api/admin/catalog/id
   ```

2 Examine the response to locate the `Link` element that contains the URL for publishing the catalog.

   This element has a `rel` attribute value of `publish` and a `type` attribute value of `application/vnd.vmware.admin.publishCatalogParams+xml`, as this example shows:

   ```
   <Link
      rel="publish"
      type="application/vnd.vmware.admin.publishCatalogParams+xml"
      href="https://vcloud.example.com/api/admin/catalog/32/action/publish" />
   ```

3 Create a `PublishCatalogParams` element that contains an `IsPublished` element with a value of `true`.

4 POST the `PublishCatalogParams` body to the catalog's `rel="publish"` URL.

Results

The catalog is shared and becomes available to members of all other organizations in the cloud.

Example: Share a Catalog With All Organizations in a Cloud

Request:

```
POST https://vcloud.example.com/api/admin/catalog/32/action/publish
Content-Type: application/vnd.vmware.admin.publishCatalogParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<PublishCatalogParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsPublished>true</IsPublished>
</PublishCatalogParams>
```

Response:

```
204 No Content
```

Share a Catalog With Specific Organizations in a Cloud

A system administrator can share a catalog to make it visible to the administrators of specific organizations in a cloud.

The owner of a catalog can share it to make it available to all organizations in the cloud. When more selective sharing is required, a system administrator can share a catalog with specific organizations if that catalog is not already shared to all organizations.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the catalog.

Use a request like this one:

```
GET https://vcloud.example.com/api/catalog/id
```

2 Examine the `Catalog` response to find the `controlAccess` links that it contains.

These links have the following form:

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.controlAccess+xml"
   href="https://vcloud.example.com/api/catalog/id/controlAccess/"/>
```

```
<Link
   rel="controlAccess"
   type="application/vnd.vmware.vcloud.controlAccess+xml"
   href="https://vcloud.example.com/api/catalog/id/action/controlAccess/"/>
```

3 Create a `ControlAccessParams` element request body that specifies the details of the update.

4 POST the `ControlAccessParams` element to the `action/controlAccess` link for the catalog.

Results

The catalog is shared with administrators of the specified organizations.

Example: Share a Catalog With Specific Organizations in a Cloud

This request updates the access controls of a catalog to share it to a single organization, giving read-only access to all users in that organization.

Request:

```
POST https://vcloud.example.com/api/catalog/32/action/controlAccess/"/>
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
       <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.organization+xml"
            href="https://vcloud.example.com/api/admin/org/355"/>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
   </AccessSettings>
</ControlAccessParams>
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<ControlAccessParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
       <AccessSetting>
         <Subject
            type="application/vnd.vmware.admin.organization+xml"
            href="https://vcloud.example.com/api/admin/org/355"/>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
   </AccessSettings>
</ControlAccessParams>
```

Publish an Existing Catalog Externally

An organization administrator in an organization that has permission to publish externally can update an existing catalog to make its contents available to external consumers on a subscription basis.

Organizations that are permitted to publish externally can enable any of their local catalogs for external publication. A catalog that is enabled for external publication becomes accessible at a URL assigned by the system, and can be protected by a password. A catalog in another instance of VMware Cloud Director can subscribe to an externally published catalog and maintain an up-to-date set of catalog items. See Synchronization.

Prerequisites

- This operation requires the rights included in the predefined Catalog Author role or an equivalent set of rights.
- Verify that your organization has permission to publish externally.

  Verify that the `OrgGeneralSettings` in the `AdminOrg` element that represents your organization has a `CanPublishExternally` element with a value of `true`.

  ```
  <CanPublishExternally>true</CanPublishExternally>
  ```
- Verify that the catalog you want to enable for external publication does not have an external subscription.

Procedure

1. In the admin view, retrieve the XML representation of the catalog.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/admin/catalog/32
   ```

2. Examine the `AdminCatalog` element to find the `publishToExternalOrganizations` link it contains.

   This link has the following form:

   ```
   <Link
      rel="publishToExternalOrganizations"
      type="application/vnd.vmware.admin.publishExternalCatalogParams+xml"
      href="https://vcloud.example.com/api/admin/catalog/32/action/
   publishToExternalOrganizations" />
   ```

3. Create a `PublishExternalCatalogParams` element.

   See Publish an Existing Catalog.

4. POST the `PublishExternalCatalogParams` element to the `publishToExternalOrganizations` URL.

Example: Publish an Existing Catalog

This request updates the catalog created in Create a Catalog to publish it externally.

Request:

```
POST https://vcloud.example.com/api/admin/catalog/32/action/publishToExternalOrganizations
Content-Type: application/vnd.vmware.admin.publishExternalCatalogparams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<PublishExternalCatalogParams
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsPublishedExternally>true</IsPublishedExternally>
   Password>Pa55w0rd</Password>
</PublishExternalCatalogParams>
```

Response:

```
204 No Content
```

Content Subscription Endpoint Specification

This document specifies the requirements for creating an endpoint that is compatible with a client using version 1 of the VMware Content Subscription Protocol (VCSP). VMware Cloud Director™ catalogs use this protocol when synchronizing content from an external subscription.

Introduction

VMware Cloud Director™ enables organizations to create catalogs that acquire their content from a subscription to an external source. The external source can be an externally published catalog hosted on another instance of VMware Cloud Director or a Web site that hosts a VCSP endpoint.

Configuration parameters for VMware Cloud Director catalogs that have external subscriptions include a VCSP endpoint URL and, if the endpoint requires authentication, a user name and password. Users periodically synchronize to update catalog contents from the subscribed endpoint. You can request synchronization for an entire catalog, or for individual catalog items. The process is similar in both cases.

1 The VCSP client makes a GET request to retrieve the endpoint descriptor. If the version in the descriptor matches the version of the catalog being synchronized, no content has changed at the endpoint, and no further action is required.

2 If the version in the descriptor is greater than the version of the catalog, the client makes a GET request to retrieve the endpoint index file.

3 To update the entire catalog, the client makes GET requests to retrieve all catalog items whose version at the endpoint is greater than their version in the catalog. To update an individual catalog item, the client makes a GET request to retrieve it only if its version number at the endpoint is greater than its version in the catalog.

The endpoint is a passive partner in this process. Its only responsibilities are to keep the version numbers up to date in the endpoint descriptor and endpoint index files, and to return the HTTP status codes and other responses expected by the VCSP client.

URLs

A VCSP endpoint URL must have a scheme of HTTP or HTTPS and a path that terminates in the name of the endpoint descriptor. A VCSP endpoint whose contents are stored on a host named `vcsp.example.com` in a directory named `/vcsp` on the host and whose descriptor file is named `descriptor.json` would have the following endpoint URL:

```
http://vcsp.example.com/vcsp/descriptor.json
```

Endpoint Descriptor

The endpoint descriptor is a file whose contents define the catalog available at the endpoint. The contents, which must be expressed in JavaScript Object Notation (media type `application/json`) as defined by RFC4627, define a JSON object, and must specify values for the following key names.

vcspVersion

An integer that specifies the version of the VCSP protocol to which this endpoint conforms.

version

An integer that specifies the version of the catalog, as described in Version Numbers.

id

The object identifier, expressed in URN format. This value uniquely identifies the catalog, persists for the life of the catalog, and must never be reused.

name

The display name of the catalog. Must be a nonempty string.

created

Time and date when the endpoint was created.

itemType

Must have value `vcsp.CatalogItem`.

itemsHref

A reference to the endpoint index file relative to the location of the endpoint descriptor. Replacing the final component of the endpoint URL with the value of the `itemsHref` key must create a valid URL.

capabilities

A JSON object that describes the capabilities of this catalog. The object must define all of the following properties:

transferIn

An array with a single member that must be the string `httpGet`.

transferOut

An array with a single member that must be the string `httpGet`.

generateIds

A Boolean value that must be set to `true`.

metadata

An array of catalog object metadata. See Metadata.

maintenanceMessage

A string indicating that the endpoint is inaccessible because it is undergoing maintenance. For example:

```
"maintenanceMessage" : "This catalog is currently in maintenance mode"
```

If the endpoint descriptor includes this key, attempts by subscribers to synchronize with this endpoint fail and display the string.

The following example shows a typical endpoint descriptor.

```
{
  "vcspVersion" : "1",
  "version" : "10",
  "id" : "urn:uuid:ccdd2c56-e54e-4883-bc0a-619baaca92a6",
  "name" : "published",
  "created" : "2012-09-14T22:17:50.807Z",
  "itemType" : "vcsp.CatalogItem",
  "itemsHref" : "items",
  "capabilities" : {
    "transferIn" : [ "httpGet" ],
    "transferOut" : [ "httpGet" ],
    "generateIds" : true
  }
   "metadata" : [ {
    "type" : "STRING",
    "domain" : "GENERAL",
    "key" : "key1",
    "value" : "value1",
    "visibility" : "READWRITE"
  }, {
    "type" : "STRING",
    "domain" : "SYSTEM",
    "key" : "key2",
    "value" : "value2",
    "visibility" : "READONLY"
  } ]
}
```

Endpoint Index

The endpoint index is a file whose contents define the set of items available in the catalog. The contents, which must be expressed in JavaScript Object Notation (media type `application/json`) as defined by RFC4627, define a JSON object, and must specify values for the following key names.

itemType

Must be `vcsp.CatalogItem`.

items

An array of zero or more `item` objects. Each `item` object must specify values for the following key names.

version

An integer that specifies the version of the item, as described in Version Numbers.

id

The object identifier of the `item`, expressed in URN format. This value uniquely identifies the `item`, persists for the life of the `item`, and must never be reused.

name

The display name of the `item`. Must be a nonempty string.

created

Time and date when the `item` was created.

type

Item type. Must have one of the following values:

- `vcsp.iso` if the `item` is an ISO image.
- `vcsp.ovf` if the `item` is an OVF package.

files

An array of `file` objects that includes all the files that represent the `item`. Each `file` object is represented as an array with three elements:

etag

An integer representing the version of the file. The value of this key must be the same for each file in the array. When any file in this array gets updated, you must increment the value of the `etag` key for all files in the array.

name

The name of the file.

hrefs

An array of pathnames to the file. Must contain a single pathname to the file from the root of the endpoint, written as a URL fragment.

properties

An array of additional properties of the `item`. This array has a single member:

selfHref

A URL to the item descriptor for this item. See Item Descriptors.

metadata

An array of catalog item metadata. See Metadata.

vms

If this item represents a vApp template, you must include an array representing the virtual machines referenced in the template.

name

The name of the virtual machine.

metadata

An array of virtual machine metadata. See Metadata.

version

An integer that specifies the version of the endpoint index, as described in Version Numbers.

The following example shows a typical endpoint index.

```
{
  "itemType" : "vcsp.CatalogItem",
  "items" : [ {
    "version" : "1",
    "id" : "urn:uuid:6dfa4596-a7c5-4d62-9a84-c559968baa26",
    "name" : "vapp-demo",
    "created" : "2012-09-17T17:59:15.161Z",
    "type" : "vcsp.ovf",
    "files" : [ {
      "etag" : "37"
      "name" : "descriptor.ovf",
      "hrefs" : [ "/vcsp/item/6dfa4596-a7c5-4d62-9a84-c559968baa26/file/descriptor.ovf" ]
    }, {
      "name" : "descriptor.mf",
      "hrefs" : [ "/vcsp/item/6dfa4596-a7c5-4d62-9a84-c559968baa26/file/descriptor.mf" ]
    }, {
      "etag" : "37"
      "name" : "vm-d5349476-aae2-4b65-bc9a-28effdc213fb-disk-0.vmdk",
      "hrefs" : [ "/vcsp/item/6dfa4596-a7c5-4d62-9a84-c559968baa26/file/vm-d5349476-aae2-4b65-
bc9a-28effdc213fb-disk-0.vmdk" ]
    } ],
    "properties" : {
    },
    "selfHref" : "/vcsp/item/6dfa4596-a7c5-4d62-9a84-c559968baa26/item.json"
  }, {
    "version" : "2",
    "id" : "urn:uuid:b63c9bbe-3614-4153-82fd-d5f4916a1327",
    "name" : "template1",
    "created" : "2012-09-14T22:18:12.858Z",
    "description" : "Added with VMware OVFTool.",
    "type" : "vcsp.ovf",
    "files" : [ {
      "name" : "descriptor.ovf",
      "hrefs" : [ "/vcsp/item/b63c9bbe-3614-4153-82fd-d5f4916a1327/file/descriptor.ovf" ]
    }, {
      "name" : "descriptor.mf",
      "hrefs" : [ "/vcsp/item/b63c9bbe-3614-4153-82fd-d5f4916a1327/file/descriptor.mf" ]
    }, {
      "name" : "vm-3fe8a95b-ccd1-4816-b4ed-d631f3e2bbf3-disk-0.vmdk",
      "hrefs" : [ "/vcsp/item/b63c9bbe-3614-4153-82fd-d5f4916a1327/file/vm-3fe8a95b-ccd1-4816-
b4ed-d631f3e2bbf3-disk-0.vmdk" ]
    } ],
    "properties" : {
    },
    "selfHref" : "/vcsp/item/b63c9bbe-3614-4153-82fd-d5f4916a1327/item.json"
  } ],
     "metadata" : []
     "vms" : [ {
      "name" : "vm1",
      "metadata" : [ ]

    }, {
      "name" : "vm2",
      "metadata" : []
    } ]
  }, {
  "version" : "10"
}
```

Metadata

The endpoint descriptor and endpoint index allow you to include object metadata that is used to create VMware Cloud Director object metadata when endpoint contents are synchronized by a subscribed catalog. For more information about the design and operation of VMware Cloud Director object metadata, see Chapter 9 Working with Object Metadata.

metadata

An array of object metadata. You may include an arbitrary number of `metadata` objects in this array, subject to the size limitations documented in VMware Cloud Director API Object Metadata Limits

type

The type of the metadata `value`. One of `STRING`, `BOOLEAN`, `DATETIME` or `NUMBER`. See VMware Cloud Director API Object Metadata Contents.

domain

The access domain of the metadata. One of `GENERAL` or `SYSTEM`. See Access Control for VMware Cloud Director API Object Metadata. WhenVMware Cloud Director object metadata is created from this endpoint, the metadata `domain` is always set to `SYSTEM`, regardless of the value you supply here.

key

An arbitrary key name.

value

The value of the key.

visibility

The visibility the metadata. One of `PRIVATE`, `READONLY` or `READWRITE`. See Access Control for VMware Cloud Director API Object Metadata. When VMware Cloud Director object metadata is created from this catalog item, the metadata `visibility` is always set to `READONLY`, regardless of the value you supply here.

Item Descriptors

Every item listed in the endpoint index must include an item descriptor file in the directory that holds all the item's files. The contents, which must be expressed in JavaScript Object Notation (media type `application/json`) as defined by RFC4627, define a JSON object, and must specify values for the following key names.

version

An integer that specifies the version of the item, as described in Version Numbers.

id

The object identifier, expressed in URN format. This value uniquely identifies the catalog, persists for the life of the catalog, and must never be reused.

name

The display name of the catalog. Must be a nonempty string.

created

Time and date when the endpoint was created.

description

A description of the item, as it will appear in the destination catalog.

type

Item type. Must have one of the following values:

- `vcsp.iso` if the `item` is an ISO image.
- `vcsp.ovf` if the `item` is an OVF package.

files

An array of `file` objects that includes all the files that represent the `item`. Each `file` object is represented as an array with two elements:

name

The name of the file.

hrefs

An array of pathnames to the file. Must contain a single pathname to the file from the root of the endpoint, written as a URL fragment.

properties

An array of additional properties of the `item`. Must be empty.

The following example shows a typical item descriptor for an OVF package.

```
{
  "version" : "1",
  "id" : "urn:uuid:b63c9bbe-3614-4153-82fd-d5f4916a1327",
  "name" : "template1",
  "created" : "2012-09-14T22:18:12.858Z",
  "description" : "Added with VMware Ovf Tool.",
  "type" : "vcsp.ovf",
  "files" : [ {
    "name" : "descriptor.ovf",
    "size" : 9985,
    "hrefs" : [ "descriptor.ovf" ]
  }, {
    "name" : "vm-3fe8a95b-ccd1-4816-b4ed-d631f3e2bbf3-disk-0.vmdk",
    "size" : 833536,
    "hrefs" : [ "vm-3fe8a95b-ccd1-4816-b4ed-d631f3e2bbf3-disk-0.vmdk" ]
  } ],
  }  "properties" : {
  }
}
```

Responses

VCSP clients retrieve catalog files, including the descriptor and the endpoint index, with GET requests. The endpoint must return one of the following responses:

- HTTP status 200 (OK) followed by the file content
- HTTP status 503 (Service unavailable) followed by a document that provides additional information. This document has media type `application/json`, and provides values for the following keys:

  status

  Status of the requested file.

  progress

  File generation progress, expressed as an integer in the range 0 to 100.

  message

  An error message describing any errors that prevented the file from being generated.

The client continues to make GET requests until it receives either a 200 response or a 503 response with a nonempty `message`.

For example, an in-progress operation could return the following response.

```
{
  "status" : "",
  "progress" : 10
}
```

An operation that has encountered an error could return the following response.

```
{
  "status" : "failed",
  "message" : "File Generation failed"
}
```

Authentication

An endpoint can require authentication. VCSP clients always present the user name `vcsp` when logging in. The endpoint can specify any password for this user, but must accept the user name `vcsp`. The user name and password are encoded as specified for Basic HTTP authentication.

Version Numbers

Version numbers appear in the endpoint descriptor and endpoint index as `version` values, which are integer values that increment monotonically. It is the responsibility of the endpoint to increment the appropriate `version` value whenever any of the following changes occur.

Changes to a catalog item

- A file in the item is added, removed, or changed.
- The name or description of the item is changed.

Changes to a catalog

- An item is added to or removed from the catalog.
- The version value of any contained catalog item changes.
- The name or description of the catalog is changed.

User and Group Administration

A newly created organization has no users or groups in it. An administrator must create or import them.

An organization can contain an arbitrary number of users and groups. Users can be created locally or managed by an external identity provider. Groups must be managed by an external identity provider. Permissions within an organization are controlled through the assignment of rights and roles to users and groups.

Local Users, Imported Users, and Groups

Local user accounts are stored in the VMware Cloud Director database and managed by the organization administrator. Local users cannot be made members of groups.

Imported users and groups must be managed at the source identity provider. If an imported user changes his password, contact information, or other account properties, those changes are not effective in VMware Cloud Director until the user is imported again. The semantics of an import operation depend on the type of the identity provider in use. See About Federation and Single Sign-On.

Modifying User or Group Metadata

An organization administrator can modify metadata such as name and description for a user or group object by creating a modified version of the `User` or `Group` element that represents the object and updating the object by making a PUT request to the object's `rel="edit"` link, supplying the modified element in the request body.

Create a Local User

An organization administrator can create user accounts that are local to the organization. Local user accounts are stored in the VMware Cloud Director database.

Every user exists within the context of an organization. An organization administrator can create a local user in an organization by POSTing a `User` element to the organization's `add` URL for users, as shown in Create a Local User.

When you create a user, you must include the `Role` and `Password` elements in the request body. The role can be a predefined role or one created by the organization administrator. For more information about retrieving a list of predefined roles, see Retrieve an Administrative View of a Cloud. For more information about creating new roles, see Create a Role in Your Organization.

Note Starting with API version 38.0, the `POST /admin/org/{id}/users` operation is deprecated.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Retrieve the list of roles in your organization.

See the `RoleReferences` element of your organization. This element contains a reference to local instance of each predefined role. It also contains references to roles created in the organization.

```
GET https://vcloud.example.com/api/admin/org/26
...
<AdminOrg ... >
   ...
   <RoleReferences>
      <RoleReference
         href="https://vcloud.example.com/api/admin/org/26/role/29971497-38da-3974-98d6-
e39bbd5b482f"
         name="Defer to Identity Provider"
         type="application/vnd.vmware.admin.role+xml" />
      <RoleReference
         href="https://vcloud.example.com/api/admin/org/26/role/2e4ad538-67f5-4d4d-
ad51-2dcd512a30f6"
         name="Console Access Only"
         type="application/vnd.vmware.admin.role+xml" />
      ...
      <RoleReference
         href="https://vcloud.example.com/api/admin/org/26/role/13a69c14-
e64c-409f-800f-0ecc470ea42d"
         name="vApp Author"
         type="application/vnd.vmware.admin.role+xml" />
   </RoleReferences>
</AdminOrg>
```

2. Create a User element that defines the user account properties.

   Include the Role and Password elements in the request body. See the request portion of Create a Local User.

3. POST the User element to the organization's add URL for users.

Results

The server creates a user account in the VMware Cloud Director database and returns an updated User element to the client.

Example: Create a Local User

This example adds the user to the organization created in Create an Organization. The user is given the predefined role vApp Author, using the role href retrieved in Step 1. The request includes an optional IsEnabled element that enables the user. If not present in the request, IsEnabled defaults to false.

The response is a User element, most of which does not appear in this example. The response includes a link that an administrator can use to edit user properties, and additional elements, such as IsDefaultCached and StoredVmQuota, whose values are inherited from the organization.

- The Password element, which must not be empty when you create a local User, is never returned.
- The ProviderType, which defines the identity provider for this user, was not specified in the request, and defaults to INTEGRATED. Local users are managed by the integrated identity provider. See About Identity Providers.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/users
Content-Type: application/vnd.vmware.admin.user+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleUser" >
   <FullName>Example User Full Name</FullName>
   <EmailAddress>user@example.com</EmailAddress>
   <IsEnabled>true</IsEnabled>
   <Role
      href="https://vcloud.example.com/api/admin/org/26/role/13a69c14-
e64c-409f-800f-0ecc470ea42d" />
   <Password>Pa55w0rd</Password>
   <GroupReferences />
</User>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.user+xml
...
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="ExampleUser"
   id="urn:vcloud:user:85"
   type="application/vnd.vmware.admin.user+xml"
   href="https://vcloud.example.com/api/admin/user/85" ... >
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/85" />
   <FullName>Example User Full Name</FullName>
   <EmailAddress>user@example.com</EmailAddress>
   <IsEnabled>true</IsEnabled>
   <ProviderType>INTEGRATED</ProviderType>
   <IsAlertEnabled>false</IsAlertEnabled>
   <IsDefaultCached>false</IsDefaultCached>
   <IsGroupRole>false</IsGroupRole>
   <StoredVmQuota>0</StoredVmQuota>
   <DeployedVmQuota>0</DeployedVmQuota>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      name="vApp Author"
      href="https://vcloud.example.com/api/admin/org/26/role/13a69c14-
e64c-409f-800f-0ecc470ea42d" />
   <GroupReferences />
</User>
```

Import a User from an LDAP Service

If an organization defines an LDAP service to use, an organization or system administrator can import user accounts from that service.

Importing a group from LDAP imports all the users in the group. See Import a Group from an LDAP Service . You can also import users individually.

Note Starting with API version 38.0, the `POST /admin/org/{id}/users` operation is deprecated.

Prerequisites

- This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.
- Verify that your organization has defined an LDAP service to use.

Procedure

1 Create a `User` element that identifies the LDAP user account to import.

   The `name` attribute of the `User` element must match the `LDAP` user name, as specified in the organization's LDAP properties. You must include the `Role` element in the request body.

2 POST the `User` element to the organization's `users` URL.

Results

The server matches the value of the `name` attribute in the request body with the value of the LDAP attribute that the organization specified in the value of the `UserName` element in the `UserAttributes` of its `OrgLdapSettings`. LDAP attributes such as `userPrincipalName` or `samAccountName` are common choices here. The server imports the user from the organization's LDAP service, and returns an updated `User` element to the client.

Example: Import a User from an LDAP Database

This example imports a user to the organization created in Create an Organization. The request includes an optional `IsEnabled` element, so the user is enabled as soon as the import is complete.

The response is a `User` element, most of which is not shown in the example. The response includes a link that an administrator can use to edit user metadata, and additional elements, such as `IsDefaultCached` and `StoredVmQuota`, inherited from organization defaults. It also includes a `NameInSource` element, which contains the user's name as stored by the LDAP server, using the server's native encoding.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/users
Content-Type: application/vnd.vmware.admin.user+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="user@example.com"
   type="application/vnd.vmware.admin.user+xml">
   <IsEnabled>true</IsEnabled>
   <IsExternal>true</IsExternal>
   <Role
      href="https://vcloud.example.com/api/admin/org/26/role/13a69c14-
e64c-409f-800f-0ecc470ea42d" />
</User>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.user+xml
...
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="user@example.com"
   id="urn:vcloud:user:85"
   type="application/vnd.vmware.admin.user+xml"
   href="https://vcloud.example.com/api/admin/user/85">
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/85" />
   <FullName>Imported User Full Name</FullName>
   <EmailAddress>user@example.com</EmailAddress>
   <IsEnabled>true</IsEnabled>
   <ProviderType>INTEGRATED</ProviderType>
   <NameInSource>\F4\D3\42\8E\6A\BC\D3</NameInSource>
   <IsAlertEnabled>false</IsAlertEnabled>
   <IsDefaultCached>false</IsDefaultCached>
   <StoredVmQuota>0</StoredVmQuota>
   <DeployedVmQuota>0</DeployedVmQuota>
   <Role
     type="application/vnd.vmware.admin.role+xml"
     name="vApp Author"
     href="https://vcloud.example.com/api/admin/org/26/role/13a69c14-
e64c-409f-800f-0ecc470ea42d" />
   <GroupReferences />
</User>
```

Import a Group from an LDAP Service

If an organization defines an LDAP service to use, an organization or system administrator can import groups from that service.

Importing a group from LDAP imports all the users in the group. You can also import users individually. See Import a User from an LDAP Service.

Prerequisites

- This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.
- Verify that your organization has defined an LDAP service to use.

Procedure

1 Create a `Group` element that identifies the LDAP group to import.

  The `name` attribute of the `Group` element must match the `LDAP` group name, as specified in the organization's LDAP properties. You must include a `Role` element in the request body. The role specified in this element is assigned to all group members during the import.

2 POST the `Group` element to the organization's `groups` URL.

Results

The server matches the value of the `name` attribute in the request body with the value of the LDAP attribute that the organization specified in the value of the `GroupName` element in the `GroupAttributes` of its `OrgLdapSettings`. The LDAP `cn` attribute is a common choice here. The server imports that group and all of its users from organization's LDAP service, and returns an updated `Group` element to the client.

Example: Import a Group from an LDAP Service

This example imports a group to the organization created in Create an Organization. The response is a `Group` element, most of which does not appear in the example. The response includes a link that an administrator can use to edit group metadata such as name and description, and a `UsersList` element that includes a `UserReference` element for each user in the group. The response also includes a `NameInSource` element, which contains the group's name as stored by the LDAP service, using its native encoding.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/groups
Content-Type: application/vnd.vmware.admin.group+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Group
   name="Engineering"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Role
        href="https://vcloud.example.com/api/admin/role/1bf4457f-a253-3cf1-b163-
f319f1a31802"/>
</Group>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.group+xml
...
<Group
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Engineering"
   id="urn:vcloud:group:44"
   type="application/vnd.vmware.admin.group+xml"
   href="https://vcloud.example.com/api/admin/group/44" ...>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      name="vApp Author"
      href="https://vcloud.example.com/api/admin/role/1bf4457f-a253-3cf1-b163-f319f1a31802" />
</Group>
```

Until the import is complete, the Group element contains only partial information. After the import is complete, the element includes a list of users and other information.

```
<Group
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Engineering"
   id="urn:vcloud:group:44"
   type="application/vnd.vmware.admin.group+xml"
   href="https://vcloud.example.com/api/admin/group/44" ...>
   <Link
```

```
      rel="edit"
      type="application/vnd.vmware.admin.group+xml"
      href="https://vcloud.example.com/api/admin/group/44" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/group/44" />
   <Description>Research and development</Description>
   <NameInSource>\C5\AF\B9\D4\9E\B5\32\40\AD\C5\E3\8E\17\4C\0D\28</NameInSource>
   <UsersList>
      <UserReference
         type="application/vnd.vmware.admin.user+xml"
         name="User-1"
         href="https://vcloud.example.com/api/admin/user/18" />
      <UserReference
         type="application/vnd.vmware.admin.user+xml"
         name="User-3"
         href="https://vcloud.example.com/api/admin/user/19" />
   </UsersList>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      name="vApp Wrangler"
      href="https://vcloud.example.com/api/admin/role/102" />
</Group>
```

Import a User or Group from an OAuth Identity Provider

If your organization defines an OAuth identity provider in its `OrgOAuthSettings`, users managed by that identity provider are created implicitly when they first log in to the organization. If you want a user or group to exist in the organization before first login, or to assign the user or group a specific role, you can import a user or group from an OAuth identity provider explicitly.

Unlike imports from an LDAP service, imports from an OAuth identity provider do not actually import information from an external database. Instead, the operation creates a mapping between a user or group defined in your organization and a user or group defined by your organization's OAuth provider. The VMware Cloud Director database stores these mappings, but does not store data retrieved from the OAuth provider.

Prerequisites

- This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.
- Verify that your organization has defined an OAuth identity provider in its `OrgOAuthSettings`.

Procedure

1. Create a `User` or `Group` element that identifies a user defined by your organization's OAuth provider.
2. Include the following line in the `User` or `Group` element.

   ```
   <ProviderType>OAUTH</ProviderType>
   ```

3 POST the element to the organization's `users` URL.

Example: Import a User from an OAuth Identity Provider

This example imports a user from an OAuth identity provider and assigns the user the `Defer to Identity Provider` role. See #unique_79.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/users
Content-Type: application/vnd.vmware.admin.user+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="user@example.com"
   type="application/vnd.vmware.admin.user+xml">
   <IsEnabled>true</IsEnabled>
   <ProviderType>OAUTH</ProviderType>
   <Role
      href="https://vcloud.example.com/api/admin/org/26/role/d26caa7b-
d8b2-395b-9615-4cd1903a65c1" />
</User>
```

The response is a `User` element, most of which is not shown in the example. The response includes a link that an administrator can use to edit user metadata, and additional elements, such as `IsDefaultCached` and `StoredVmQuota`, inherited from organization defaults. It also includes a `NameInSource` element, which contains the user's name in the encoding used by the identity provider.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.user+xml
...
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="user@example.com"
   id="urn:vcloud:user:85"
   type="application/vnd.vmware.admin.user+xml"
   href="https://vcloud.example.com/api/admin/user/85">
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/85" />
   <FullName>Imported User Full Name</FullName>
   <EmailAddress>user@example.com</EmailAddress>
   <IsEnabled>true</IsEnabled>
   <ProviderType>OAUTH</ProviderType>
   <NameInSource>\F4\D3\42\8E\6A\BC\D3</NameInSource>
   <IsAlertEnabled>false</IsAlertEnabled>
   <IsDefaultCached>false</IsDefaultCached>
   <StoredVmQuota>0</StoredVmQuota>
   <DeployedVmQuota>0</DeployedVmQuota>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      name="Defer to Identity Provider"
      href="https://vcloud.example.com/api/admin/org/26/role/d26caa7b-
d8b2-395b-9615-4cd1903a65c1" />
   <GroupReferences />
</User>
```

Import a User or Group from a SAML Identity Provider

If your organization defines a SAML identity provider in its `OrgFederationSettings`, you must import users individually or as members of groups from the identity provider before they can log in to the organization.

Unlike imports from an LDAP service, imports from a SAML identity provider do not import information from an external database. Instead, the operation creates a mapping between a user or group name in your organization's database and a user or group name that your organization's SAML provider defines. The VMware Cloud Director database stores these mappings, but does not store data retrieved from the SAML provider.

When you import a user from a SAML identity provider, you must include the domain name, such as `user@domain.com`. When you import a group from a SAML identity provider, you must use its fully distinguished name.

When a user login presents a SAML token to the organization, user and group names in the token are evaluated using the mappings established by the import operation. If you defined custom user and group attributes, the evaluation process uses the values of the `UserNameAttributeName` and `GroupNameAttributeName` elements from the `SamlAttributeMapping` element in the organization `OrgFederationSettings`. The evaluation process includes the following:

- If the SAML token includes an attribute matching the value of the `UserNameAttributeName` element or an attribute named `UserName`, tries to match the value of that attribute to the value of the `name` attribute of the `User`. If the SAML token does not include such an attribute, tries to match the value of the `NameId` element to the value of the `name` attribute of the `User`.
- If the SAML token includes an attribute matching the value of `GroupNameAttributeName` or an attribute named `Groups`, assumes that the value of that attribute is a list of group names, and tries to match each value in the list to the value of the `name` attribute of a `Group` in the organization. If the SAML token does not include such an attribute, assumes that the user is not a member of any group.

Prerequisites

- This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.
- Verify that your organization has defined a SAML identity provider in its `OrgFederationSettings`. See Configuring and Managing Federation with SAML.

Procedure

1 Create a `User` or `Group` element that identifies a user or group that your organization's SAML provider defines.

2 Include the following line in the `User` or `Group` element.

```
<ProviderType>SAML</ProviderType>
```

3 POST the element to the organization's `users` or `groups` URL.

Example: Import a User from a SAML Identity Provider

This example imports a user from the SAML identity provider that the organization defines. This example is identical to the one shown in Import a User from an LDAP Database, but includes a `ProviderType` element that specifies the source as the organization's SAML identity provider. This example also omits the `IsExternal` element, which is required when importing from LDAP but is ignored when importing from SAML.

The response is a `User` element, most of which is not shown in the example. The response includes a link that an administrator can use to edit user metadata, and additional elements, such as `IsDefaultCached` and `StoredVmQuota`, inherited from organization defaults. It also includes a `NameInSource` element, which contains the user's name in the encoding that the identity provider uses.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/users
Content-Type: application/vnd.vmware.admin.user+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="user@example.com"
   type="application/vnd.vmware.admin.user+xml">
   <IsEnabled>true</IsEnabled>
   <ProviderType>SAML</ProviderType>
   <Role
      href="https://vcloud.example.com/api/admin/org/26/role/13a69c14-
e64c-409f-800f-0ecc470ea42d" />
</User>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.user+xml
...
<User
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="user@example.com"
   id="urn:vcloud:user:85"
   type="application/vnd.vmware.admin.user+xml"
   href="https://vcloud.example.com/api/admin/user/85">
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.user+xml"
      href="https://vcloud.example.com/api/admin/user/85" />
   <FullName>Imported User Full Name</FullName>
   <EmailAddress>user@example.com</EmailAddress>
   <IsEnabled>true</IsEnabled>
   <ProviderType>SAML</ProviderType>
   <NameInSource>\F4\D3\42\8E\6A\BC\D3</NameInSource>
   <IsAlertEnabled>false</IsAlertEnabled>
   <IsDefaultCached>false</IsDefaultCached>
   <StoredVmQuota>0</StoredVmQuota>
   <DeployedVmQuota>0</DeployedVmQuota>
   <Role
      type="application/vnd.vmware.admin.role+xml"
      name="vApp Author"
      href="https://vcloud.example.com/api/admin/org/26/role/13a69c14-
e64c-409f-800f-0ecc470ea42d" />
   <GroupReferences />
</User>
```

Take Control of a User's Objects

An administrator can take control of a user's vApps, media, and catalogs.

When an administrator retrieves a `User` element, it contains a link that the administrator can use to take ownership of that user's vApps, media, and catalogs. This action is typically required when a user leaves the organization and an administrator must transfer ownership of that user's objects to other users.

Note Starting with API version 38.0, the following endpoints are deprecated.

API Endpoint	Description
`GET /admin/user/{id}`	Retrieve a user.
`PUT /admin/user/{id}`	Update a user.
`DELETE /admin/user/{id}`	Delete a user.
`POST /admin/user/{id}/action/takeOwnership`	Transfer ownership of this user's vApps, media, and catalogs to the called.
`GET /admin/users/query`	Retrieves a list of users for the organization that the organization administrator belongs to by using REST API general QueryHandler.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Retrieve the XML representation of the user.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/user/85
```

2 Examine the response to find the link where `rel="takeOwnership"`.

```
<User
   ...
   href="https://vcloud.example.com/api/admin/user/85" ... >
   ...
   <Link
      rel="takeOwnership"
      href="https://vcloud.example.com/api/admin/user/85/action/takeOwnership"/>
   ...
</User>
```

3 Make a POST request to the URL in the `href` value of that link.

```
POST  https://vcloud.example.com/api/admin/user/85/action/takeOwnership
```

The response is empty (204 No Content).

Results

All vApps, media, and catalogs that belonged to the user now belong to the administrator who made the request. The administrator can then change the ownership as needed. See View or Change the Owner of an Object.

About Federation and Single Sign-On

An identity provider is a service that manages user and group identities. VMware Cloud Director organizations that use the same identity provider are said to be federated.

An organization can define an identity provider that it shares with other applications or enterprises. Users authenticate to the identity provider to obtain a token that they can then use to log in to the organization. Such a strategy can enable an enterprise to provide access to multiple, unrelated services, including VMware Cloud Director, with a single set of credentials, an arrangement often referred to as single sign-on.

VMware Cloud Director includes a Multisite capability that extends the advantages of federation by enabling administrators to associate organizations with each other so that a user authenticated to one organization is also authenticated to all organizations that it is associated with. For organizations, federation (sharing of an IDP) is prerequisite to association. See Configuring and Managing Multisite Deployments for more information about associating sites and organizations.

About Identity Providers

VMware Cloud Director supports the following kinds of identity providers:

OAuth

An organization can define an external identity provider that supports OAuth authentication, as defined in RFC 6749 (http://tools.ietf.org/html/rfc6749). All organizations that participate in an OAuth-based federated identity scheme must include an `OrgOAuthSettings` element whose public key, `IssuerId` and `OAuthKeyConfigurations` were retrieved from the same identity provider.

SAML

An organization can define an external identity provider that supports the Security Assertion Markup Language (SAML) 2.0 standard. All organizations participating in a SAML-based federated identity scheme must include an `OrgFederationSettings` element that contains SAML metadata retrieved from the same identity provider.

Integrated

The integrated identity provider is a VMware Cloud Director service that authenticates users who are created locally or imported from LDAP. All organizations that participate in an LDAP-based federated identity scheme must include an `OrgLdapSettings` element that specifies shared configuration parameters.

The XML representation of a `User` can include an `IdentityProvider` element that specifies `INTEGRATED`, `OAUTH`, or `SAML`. If the element is missing or empty, a value of `INTEGRATED` is assumed.

The XML representation of a `Group` can include a `ProviderType` element that specifies `INTEGRATED` or `SAML`. If the element is missing or empty, a value of `INTEGRATED` is assumed.

Configuring and Managing Federation with OAuth

An organization can define an external identity provider that supports OAuth 2.0 authentication, as defined in RFC 6749. All organizations that participate in an OAuth-based federated identity scheme must include an `OrgOAuthSettings` element whose `IssuerId` and `OAuthKeyConfigurations` were retrieved from the same identity provider.

Note Starting with VMware Cloud Director API v. 31.0, OAuth login is based on OpenID Connect.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Select an OAuth service that supports OAuth 2.0. To enable your organization to use this service, you must obtain the service's public key and related information that the `OrgOAuthSettings` element of your organization's `Settings` element requires.

Procedure

1 Retrieve your organization's `OrgOAuthSettings`.

   Use a request similar to:

   ```
   GET https://vcloud.example.com/api/admin/org/40/settings/oauth
   ```

2 To add your identity provider's OAuth metadata, modify the retrieved `OrgOAuthSettings` element.

   This metadata includes the service's public key, issuer identifier, a URL to which you can make an authentication request, at least one key configuration, access token endpoint, user authorization endpoint, and user information endpoint. See Update Organization OAuth Settings.

   To retrieve user information, you must use either the `UserInfoEndpoint` or the `ScimEndpoint` element. If you want to retrieve group information, you must use the `ScimEndpoint` element. When using the `UserInfoEndpoint` element, optionally, you can customize the key names by using the `OIDCAttributeMapping` element.

   Note Starting with VMware Cloud Director API version 38.0, the `ScimEndpoint` field is deprecated.

   Note If you are using VMware Identity Manager® as your OAuth identity provider, you must use the `ScimEndpoint` element.

3 Update the `OrgOAuthSettings` with your modifications.

   a Find the `Link` element in the settings element where `rel="edit"`.

   b Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body. See the request portion of Update Organization Federation Settings.

Example: Update Organization OAuth Settings

This example updates the `OrgOAuthSettings` of an organization whose URL is `https://vcloud.example.com/api/admin/org/40/`. The update adds information retrieved from an identity provider, and enables OAuth federation by setting `Enabled` to `true`.

Request:

```
PUT https://vcloud.example.com/api/admin/org/40/settings/oauth
Content-Type: application/vnd.vmware.admin.organizationOAuthSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgOAuthSettings xmlns="http://www.vmware.com/vcloud/v1.5">
    <IssuerId>http://IDP.example.com/oauth2</IssuerId>
    <OAuthKeyConfigurations>
        <OAuthKeyConfiguration>
```

```
            <KeyId>f2842a04-a172-407d-bac3-20f4a175af3e</KeyId>
            <Algorithm>RSA</Algorithm>
            <Key>-----BEGIN PUBLIC KEY-----
         MIIBIjANBgk...
         ...
         -----END PUBLIC KEY-----</Key>
        </OAuthKeyConfiguration>
    </OAuthKeyConfigurations>
    <Enabled>true</Enabled>
    <ClientId>clientId</ClientId>
    <ClientSecret>clientSecret</ClientSecret>
    <UserAuthorizationEndpoint>https://IDP.example.com/oauth2/authorize</
UserAuthorizationEndpoint>
    <AccessTokenEndpoint>https://IDP.example.com/oauth2/access_token</AccessTokenEndpoint>
    <UserInfoEndpoint>https://IDP.example.com/oauth2/userinfo</UserInfoEndpoint>
    <Scope>openid</Scope>
    <Scope>email</Scope>
    <Scope>profile</Scope>
        <OIDCAttributeMapping>
     <SubjectAttributeName>sub</SubjectAttributeName>
     <EmailAttributeName>email</EmailAttributeName>
     <FirstNameAttributeName>givenname</FirstNameAttributeName>
     <LastNameAttributeName>surname</LastNameAttributeName>
     <GroupsAttributeName>groups</GroupsAttributeName>
     <RolesAttributeName>roles</RolesAttributeName>
    </OIDCAttributeMapping>
    <MaxClockSkew>60</MaxClockSkew>
</OrgOAuthSettings>
```

The response contains information extracted from the request, and includes `Link` elements that the server creates.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.organizationOAuthSettings+xml
...
<OrgOAuthSettings href="https://vcloud.example.com/api/admin/org/40/settings/oauth"
         type="application/vnd.vmware.admin.organizationOAuthSettings+xml">
         <Link
            rel="up"
            href="https://vcloud.example.com/api/admin/org/40/settings"
            type="application/vnd.vmware.admin.orgSettings+xml" />
         <Link
            rel="edit"
            href="https://vcloud.example.com/api/admin/org/40/settings/oauth"
            type="application/vnd.vmware.admin.organizationOAuthSettings+xml" />
    <IssuerId>http://IDP.example.com/oauth2</IssuerId>
    <OAuthKeyConfigurations>
        <OAuthKeyConfiguration>
            <KeyId>f2842a04-a172-407d-bac3-20f4a175af3e</KeyId>
            <Algorithm>RSA</Algorithm>
            <Key>-----BEGIN PUBLIC KEY-----
         MIIBIjANBgk...
         ...
         -----END PUBLIC KEY-----</Key>
        </OAuthKeyConfiguration>
    </OAuthKeyConfigurations>
    <Enabled>true</Enabled>
    <ClientId>clientId</ClientId>
    <ClientSecret>clientSecret</ClientSecret>
    <UserAuthorizationEndpoint>https://IDP.example.com/oauth2/authorize</
UserAuthorizationEndpoint>
    <AccessTokenEndpoint>https://IDP.example.com/oauth2/access_token</AccessTokenEndpoint>
    <UserInfoEndpoint>https://IDP.example.com/oauth2/userinfo</UserInfoEndpoint>
    <Scope>openid</Scope>
    <Scope>email</Scope>
    <Scope>profile</Scope>
        <OIDCAttributeMapping>
     <SubjectAttributeName>sub</SubjectAttributeName>
     <EmailAttributeName>email</EmailAttributeName>
     <FirstNameAttributeName>givenname</FirstNameAttributeName>
     <LastNameAttributeName>surname</LastNameAttributeName>
     <GroupsAttributeName>groups</GroupsAttributeName>
     <RolesAttributeName>roles</RolesAttributeName>
    </OIDCAttributeMapping>
    <MaxClockSkew>60</MaxClockSkew>
</OrgOAuthSettings>
```

Enable PKCE in VMware Cloud Director

Starting with version 10.5, VMware Cloud Director supports Proof Key for Code Exchange (PKCE).

PKCE is an extension to the OAuth 2.0 Authorization Code flow that is used to prevent CSRF and authorization code injection attacks. For more information, see Proof Key for Code Exchange in the OAuth 2.0 documentation.

For more details on using VMware Cloud Director API for OAuth configuration, see VMware Cloud Director API and Configuring and Managing Federation with OAuth.

Procedure

1 Run the request to retrieve your organization's settings.

```
GET https://vcloud.example.com/api/admin/org/organization_id/settings/oauth
```

The response contains the OAuth settings for your organization.

2 Under `OrgOAuthSettings`, make the following changes.

a Modify the `usePkce` element to `true`.

b (Optional) If you want to send client credentials in the authorization header of API requests, modify the `sendClientCredentialsAsAuthorizationHeader` element to `true`.

3 To update your OAuth settings with your modifications, run a PUT request.

```
PUT https://vcloud.example.com/api/admin/org/organization_id/settings/oauth
```

In the body of the request, include the modified elements of your OAuth settings.

Configuring and Managing Federation with SAML

An organization can define an external identity provider that supports the Security Assertion Markup Language (SAML) 2.0 standard. All organizations participating in a SAML-based federated identity scheme must include an `OrgFederationSettings` element that contains SAML metadata retrieved from the same identity provider.

When you create an organization, it receives a self-signed certificate for use when establishing trust with an identity provider. This certificate expires after one year. You can regenerate this certificate by making a request of the following form.

```
POST https://vcloud.example.com/api/admin/org/id/settings/federation/action/
regenerateCertificate
```

You can retrieve this certificate with a request of the following form, where *name* is the name of the organization.

```
GET https://vcloud.example.com/cloud/org/name/saml/metadata/alias/vcd
```

You can make this request without authentication.

You can also add your own certificate chain and private key by including a `SamlSPKeyAndCertificate` element in your `OrgFederationSettings` update.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Download the appropriate SAML metadata in XML format from your identity provider. The SAML metadata must provide mappings for the default or custom user attributes. The following example XML fragment shows mappings to the default user attributes. Optionally, you can define custom user attributes by using the `SamlAttributeMapping` element in your organization `OrgFederationSettings` as shown in Update Organization Federation Settings.

```
<saml:Attribute
   FriendlyName="Groups"
   Name="http://rsa.com/schemas/attr-names/2009/01/GroupIdentity"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
<saml:Attribute
   FriendlyName="givenName"
   Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
<saml:Attribute
   FriendlyName="surname"
   Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
<saml:Attribute
   FriendlyName="Subject Type"
   Name="http://vmware.com/schemas/attr-names/2011/07/isSolution"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
<saml:Attribute
   FriendlyName="userPrincipalName"
   Name="http://schemas.xmlsoap.org/claims/UPN"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
<saml:Attribute
   FriendlyName="email"
   Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
```

Procedure

1. Retrieve the `OrgFederationSettings`.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/admin/org/40/settings/federation
   ```

2. To add your identity provider's SAML metadata as the value of the `SAMLMetadata` element, modify the retrieved `OrgFederationSettings` element.

   XML entities must be encoded, as shown in Update Organization Federation Settings

3. Update the `OrgFederationSettings` with your modifications.

   a. Find the `Link` element in the settings element where `rel="edit"`.

   b. Make a PUT request to the URL in that link's `href` attribute value, and supply the modified section as the request body. See the request portion of Update Organization Federation Settings.

Example: Update Organization Federation Settings

This example updates the `OrgFederationSettings` of an organization whose URL is `https://vcloud.example.com/api/admin/org/40/`. The update adds SAML metadata retrieved from an identity provider, and enables federation by setting `Enabled` to `true`. Only a subset of the SAML metadata appears.

Request:

```
PUT https://vcloud.example.com/api/admin/org/40/settings/federation
Content-Type: application/vnd.vmware.admin.organizationFederationSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgFederationSettings
   type="application/vnd.vmware.admin.organizationFederationSettings+xml">
   <SAMLMetadata>
      &lt;EntitiesDescriptor
      xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" .... &gt;
      ...
      &lt;/EntitiesDescriptor&gt;</SAMLMetadata>
   <SamlAttributeMapping>
     <EmailAttributeName>email</EmailAttributeName>
```

```
     <UserNameAttributeName>userPrincipalName</UserNameAttributeName>
     <FirstNameAttributeName>givenName</FirstNameAttributeName>
     <SurnameAttributeName>surname</SurnameAttributeName>
     <FullNameAttributeName>name</FullNameAttributeName>
     <GroupAttributeName>Groups</GroupAttributeName>
     <RoleAttributeName>Roles</RoleAttributeName>
   </SamlAttributeMapping>
   <Enabled>true</Enabled>
</OrgFederationSettings>
```

Note To update or remove `OrgFederationSettings` after you specify a SAML identity provider, you must retrieve the `SAMLMetadata` element from a `GET` request to the endpoint and include the `Enabled`, `SamlSPEntityId`, and `SamlAttributeMapping` elements in the update request. If you do not, the request fails without changing the `OrgFederationSettings`.

The response contains information extracted from the request, and includes `Link` elements that the server creates.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.organizationFederationSettings+xml
...
<OrgFederationSettings ...
   type="application/vnd.vmware.admin.organizationFederationSettings+xml">
   <Link
      rel="up"
      href="https://vcloud.example.com/api/admin/org/40/settings"
      type="application/vnd.vmware.admin.organization+xml"/>
   <Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/org/40/settings/federation"
      type="application/vnd.vmware.admin.organizationFederationSettings+xml"/>
   <Link
      rel="federation:regenerateFederationCertificate"
      href="https://vcloud.example.com/api/admin/org/40/settings/federation/action/
regenerateFederationCertificate"/>
   <SAMLMetadata>&lt;EntitiesDescriptor
      xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" .... &gt;
      ...
      &lt;/EntitiesDescriptor&gt; </SAMLMetadata>
   <SamlAttributeMapping>
     <EmailAttributeName>email</EmailAttributeName>
     <UserNameAttributeName>userPrincipalName</UserNameAttributeName>
     <FirstNameAttributeName>givenName</FirstNameAttributeName>
     <SurnameAttributeName>surname</SurnameAttributeName>
     <FullNameAttributeName>name</FullNameAttributeName>
     <GroupAttributeName>Groups</GroupAttributeName>
     <RoleAttributeName>Roles</RoleAttributeName>
   </SamlAttributeMapping><Enabled>true</Enabled>
</OrgFederationSettings>
```

Configuring and Managing Federation with LDAP

An organization can define an LDAP configuration that it shares with other organizations. This shared configuration can support federation using LDAP as a directory service, an authentication service, or both.

When several organizations use the same LDAP service as their source for imported users and groups, they enable a simple model of federation in which users in all the participating organizations can be managed by a single LDAP service. In this kind of configuration, user credentials are imported into the VMware Cloud Director database, and VMware Cloud Director is responsible for authenticating users.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Retrieve the organization LDAP settings.

  Use a request like this one:

  ```
  GET https://vcloud.example.com/api/admin/org/26/settings/ldap
  ```

  The response is an OrgLdapSettings element.

2 Modify the retrieved OrgLdapSettings element.

  You can set the value of OrgLdapMode to SYSTEM to specify that this organization uses the system LDAP configuration. When you do this, you can also specify a CustomUsersOu value so that only users in a specific LDAP organizational unit can be imported into this organization. See Update Organization LDAP Settings.

  To configure an LDAP service for the exclusive use of this organization, set the value of OrgLdapMode to CUSTOM and include a CustomOrgLdapSettings element in the modified OrgLdapSettings.

3 Update the OrgLdapSettings with your modifications.

  Find the Link element in the settings element where rel="edit". Make a PUT request to the URL in that link's href attribute value, and supply the modified section as the request body. See Update Organization LDAP Settings.

Example: Update Organization LDAP Settings

This example updates the OrgLdapSettings of the organization created in Create an Organization. The update sets the value of the CustomUsersOu element to specify that only LDAP users whose OU attribute has a value of Finance are imported into this organization.

Request:

```
PUT https://vcloud.example.com/api/admin/org/26/settings/ldap
Content-Type: application/vnd.vmware.admin.organizationLdapSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgLdapSettings ... >
   <OrgLdapMode>SYSTEM</OrgLdapMode>
   <CustomUsersOu>OU=Finance</CustomUsersOu>
</OrgLdapSettings>
```

Managing Rights and Roles

A right is the fundamental unit of access control in VMware Cloud Director. A role associates a role name with a set of rights. Each organization can have different rights and roles.

VMware Cloud Director uses roles and their associated rights to determine whether a user or group is authorized to perform an operation. Many of the procedures documented in the VMware Cloud Director guides include a prerequisite role. These prerequisites assume that the named role is the unmodified predefined role or a role that includes an equivalent set of rights.

System administrators can use rights bundles and global tenant roles to manage the rights and roles that are available to each organization.

After you install VMware Cloud Director, the system contains only the System Rights Bundle, which includes all rights that are available in the system. The System Rights Bundle is not published to any organization. The system also contains built-in global tenant roles that are published to all organizations. For information about the predefined roles, see #unique_79.

After you upgrade VMware Cloud Director from version 9.1 or earlier, in addition to the System Rights Bundle, the system contains a Legacy Rights Bundle for each existing organization. Each Legacy Rights Bundle includes the rights that are available in the associated organization at the time of the upgrade and is published only to this organization.

Note To begin using the rights bundles model for an existing organization, you must delete the corresponding Legacy Rights Bundle.

If you upgraded VMware Cloud Director from version 9.1 or earlier, the existing role templates are published to all organizations as global tenant roles, and the existing roles that are unlinked from role templates are available as tenant-specific roles to their organizations.

Important VMware Cloud Director introduces OpenAPIs for managing rights and roles. For information about the VMware Cloud Director OpenAPI, see *Getting Started with VMware Cloud Director OpenAPI* at https://code.vmware.com.

Rights Terminology

Right

Each right provides view or manage access to a particular object type in VMware Cloud Director. Rights belong to different categories depending on the objects to which they relate, for example, vApp, Catalog, Organization, and so on. The Provider organization contains all rights available in the system. The system administrator defines the rights that are available to each organization. You cannot create or modify the rights included in VMware Cloud Director.

Note You can create and modify rights associated with extension services, but not those associated with VMware Cloud Director. See Create a Service-Specific Right

Rights Bundle

System administrators can use rights bundles to manage the rights that are available to each organization. A rights bundle is a set of rights that the system administrator can publish to one or more organizations. The system administrator can create and publish rights bundles that correspond to tiers of service, separately monetizable functionality, or any other arbitrary rights grouping. Only system administrators can view and manage the rights bundles. You can publish multiple bundles to the same organization.

Organization Rights

Organization rights are the full set of rights that are available to an organization. Organization rights can comprise multiple rights bundles, but the organization administrators and users see a flat set of rights that they can use to create and modify tenant-specific roles.

Roles Terminology

Role

A role is a set of rights that is assignable to one or more users and groups. When you create or import a user or group, you must assign it a role.

Provider Roles

Provider roles are the set of roles that are available only to the Provider organization. Provider roles can be assigned only to Provider users. System administrators can create custom provider roles.

Tenant Roles

Tenant roles are the set of roles available to an organization.

System administrators can create and edit global tenant roles and publish them to one or more organizations. Global tenant roles can be assigned to tenant users in the organizations to which they are published. Organization administrators cannot edit global tenant roles.

Note Tenant users can use only those rights from their roles that are published to their organizations.

Tenant-Specific Roles

Organization administrators can create and edit tenant-specific roles, which are local to their organizations. Tenant-specific roles can be assigned only to tenant users in the organization to which they belong. Tenant-specific roles can contain a subset of the organization rights only.

For information about managing tenant-specific roles, see *VMware Cloud Director Tenant Guide*.

New Rights in This Release

VMware Cloud Director 10.0 introduces new rights, which you might want to add to any existing global roles that you published to your tenants.

Right	Description	Default Role
Gateway Services: Configure ECMP Routing	Allows you to configure ECMP routing on edge gateways.	**System administrator** and **Organization administrator**
Gateway Services: Configure Edge Gateway DNS	Allows you to configure the DNS settings on an edge gateway.	**System administrator**, **Organization administrator**, **Catalog Author**, **vApp Author**, and **vApp User**, if the user has at least one of the **Gateway Services: Configure** rights.
Gateway Services: View Edge Gateway DNS	Allows you to view the DNS settings on an edge gateway.	**System administrator**, **Organization administrator**, **Catalog Author**, **vApp Author**, and **vApp User**, if the user has at least one of the **Gateway Services: Configure** rights.
Organization: View API Explorer	Allows you to access the API Explorer endpoint.	**System administrator**
SDDC: Manage Token	Allows you to create and delete your tokens. **System administrators** must assign the right to any user by using proxies so that the users can renew their expired tokens.	**System Administrator**
SDDC: Manage Token All	Allows you to view and delete tokens of all users within your organization. **System administrators** can assign the right to **organization administrators** so that they can revoke tokens of users in their organization.	**System Administrator**

For information about managing rights and roles, see the *VMware Cloud Director Service Provider Admin Guide*.

Create a Role in Your Organization

A system administrator can use the VMware Cloud Director Service Provider Admin Portal or the VMware Cloud Director API to create or update role objects in any organization in the system.

Organization administrators can use the VMware Cloud Director API to create or update role objects in the organizations they administer.

Role and right objects are local to an organization. An organization is initially granted a set of rights derived from the rights contained in the predefined roles, and includes a copy of each predefined role. A system administrator can grant additional rights to an organization. See Edit Organization Rights.

Organization administrators can create or update roles in organizations they administer by aggregating a set of rights in a `Role` element and POSTing it to the organization's `add` URL for roles. Roles created in this way are local to a specific organization.

Important An organization administrator cannot modify a predefined role or create a new role that has the same name as a predefined role.

Prerequisites

This operation requires the rights included in the predefined Organization Administrator role or an equivalent set of rights.

Procedure

1 Retrieve the set of rights available to your organization.

   To get the `RightReference` objects that populate the `Role`, use a request like this one:

   ```
   GET https://vcloud.example.com/api/admin/org/id/rights
   ```

   The `OrgRights` element returned by this request includes a `RightReference` element for each right granted to the organization by the system administrator.

2 Create a `Role` element that defines the role with a name and a set of rights.

   See the request portion of Create a Role.

3 POST the `Role` element to the organization's `add` URL for `roles`.

Results

The system creates the role in your organization and returns its representation, a `Role` element, in the response.

Example: Create a Role

This example adds a role named `vAppWrangler` to the organization with id `21`. The new role is created in this organization, but not in any other organizations in the system. You must be a system administrator or an administrator of this organization to make this request. The rights associated with this new role are less comprehensive than those associated with the built-in `vApp Author` role, but still include rights to perform many common vApp operations. This example uses `href` attributes that contain actual UUID values for specific rights, since these are invariant across VMware Cloud Director installations and releases.

Request:

```
POST https://vcloud.example.com/api/admin/org/21/roles
Accept: application/*;version=29.0
Content-Type: application/vnd.vmware.admin.role+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Role
   name="vAppWrangler"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Create and manage vApps</Description>
   <RightReferences>
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Copy"
         href="https://vcloud.example.com/api/admin/org/21/right/4965b0e7-9ed8-371d-8b08-
fc716d20bf4b" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Create / Reconfigure"
         href="https://vcloud.example.com/api/admin/org/21/right/2dc8abec-2e0d-3789-a5f9-
ce0453160b53" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Delete"
         href="https://vcloud.example.com/api/admin/org/21/right/df05c07f-c537-3777-8d9b-
a9cfe8d49014" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit Properties"
         href="https://vcloud.example.com/api/admin/org/21/right/c2a29357-1b2a-3f9d-9cd6-
de3d525d49f3" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM CPU"
         href="https://vcloud.example.com/api/admin/org/21/right/729a3828-8b63-31b2-88db-
f56612a06722" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM Hard Disk"
         href="https://vcloud.example.com/api/admin/org/21/right/cd02b5f8-c54a-334a-
b782-5d31a1d77d85" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM Memory"
         href="https://vcloud.example.com/api/admin/org/21/right/c6c827dc-
fc42-33a8-844f-8ab5a91f8a6c" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM Network"
         href="https://vcloud.example.com/api/admin/org/21/right/f24fffde-
f953-3976-9f2b-8b355b25881d" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Edit VM Properties"
         href="https://vcloud.example.com/api/admin/org/21/right/
5250ab79-8f50-33f9-8af5-015cb39c380b" />
      <RightReference
         type="application/vnd.vmware.admin.right+xml"
         name="vApp: Power Operations"
         href="https://vcloud.example.com/api/admin/org/21/right/580860cd-55bc-322d-
ac39-4f9d8e3e1cd2" />
   </RightReferences>
</Role>
```

The response is a `Role` element, most of which does not appear in this excerpt. The response includes links that an administrator can use to edit or remove the role.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.role+xml
...
<Role
   name="vAppWrangler"
   ...
   href="https://vcloud.example.com/api/admin/org/21/role/102" ...>
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.role+xml"
      href="https://vcloud.example.com/api/admin/org/21/role/102"/>
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/org/21/role/102"/>
   <Description>Create and manage vApps</Description>
   <RightReferences>
      ...
   </RightReferences>
</Role>
```

Managing and Monitoring a Cloud 8

The VMware Cloud Director API includes extensions that support operations on the vSphere platform, which provides resources to VMware Cloud Director. A system administrator can use these extensions to retrieve or update the configuration of a cloud, add or remove vSphere resources, and import vApps and media from vCenter.

Only the system administrator can perform vSphere platform operations. Before you attempt these operations, log in to the System organization with the user name and password of the system administrator account that was created when VMware Cloud Director was installed. See Administrator Credentials and Privileges .

For a full list of the VMware Cloud Director API extension operations, see the *VMware Cloud Director API Schema Reference* at https://code.vmware.com.

Read the following topics next:

- Retrieve or Update System Settings
- Register an NSX Manager Instance
- Finding Available vCenter Resources
- Create an Organization
- Create a Provider VDC
- Create an External Network
- Create a Network Pool
- Add a VDC to an Organization
- Managing VM-Host Affinity Rules
- Creating and Managing VDC Templates
- Import a Virtual Machine from vCenter
- Relocate a Virtual Machine to a Different Datastore
- Migrate Tenant Storage
- Retrieve the vSphere URL of an Object
- Configuring and Managing Multisite Deployments

- Managing Dedicated vCenter Server Instances

Retrieve or Update System Settings

System settings establish systemwide behaviors and default values.

The `SystemSettings` element includes all system settings for this VMware Cloud Director server group. The element also includes links that allow you to retrieve these subsidiary elements, which define specific categories of settings.

GeneralSettings

Specify service URLs, timeouts, truststore details, and similar global properties of the server group.

NotificationsSettings

Control the VMware Cloud Director AMQP notifications service.

LdapSettings

Specify details of the system LDAP directory service.

AmqpSettings

Specify credentials and connection information for the AMQP broker that handles notifications and blocking task messages.

EmailSettings

Define configuration and connection parameters for the system default email service, and specifies properties of email alerts that the system sends.

License

System license serial number and related settings.

BrandingSettings

Customize the branding of the VMware Cloud Director HTML5 UI and some of the links that appear on the VMware Cloud Director Home login screen.

BlockingTaskSettings

Control the behavior of blocking tasks and enable blocking for specific operations.

PasswordPolicySettings

Specify default policies to be followed when a user in any organization enters an incorrect password. Organization administrators can override this default for their organizations.

CPoM

Configure the dedicated vCenter Server settings. You can configure the maximum number of active proxies for the system and for a user. You can configure the timeout value in seconds for testing a connection to an SDDC proxied host. You can allow the attached vCenter Server instances to be both tenant-scoped and provider-scoped, that is one vCenter Server instance can back both a provider VDC and a dedicated vCenter Server instance. You can allow proxying over insecure HTTP in addition to HTTPS.

LookupServiceSettings

Specify the vSphere Lookup Service URL. Certain VMware Cloud Director services need this URL to authenticate with vSphere.

CatalogSettings

Specify the schedule for background synchronization of catalogs that have an external subscription. These settings apply to all organizations in the system.

You can retrieve the entire `SystemSettings` element to view all of these settings. To update an individual subsection, retrieve it with a GET request, modify it, and update it with a PUT request.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the `SystemSettings` element.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension/settings
```

2 Examine the response to locate the `Link` elements that you can use to retrieve an individual subsection.

These links have a `rel` attribute value of `down`.

3 Use the link to retrieve the subsection.

Make a GET request to the `href` value of the link.

4 (Optional) Modify the retrieved subsection.

Subsections that you can modify include a link where `rel="edit"`.

5 (Optional) To update the subsection, PUT the modified subsection to the `href` of the link described in Step 4.

Register an NSX Manager Instance

You can register an NSX Manager instance with your cloud so that you can create Provider VDCs backed by this NSX Manager instance and import logical switches as organization networks.

If you want to use a logical switch from an NSX Manager instance as an organization network in your cloud, you must first register an NSX Manager instance, then create provider and organization VDCs backed by this NSX Manager instance.

Prerequisites

- This operation is restricted to system administrators.
- Verify that you know the IP address and administrator credentials for the NSX Manager instance.

Procedure

1 Retrieve the XML representation of the cloud extension objects and operations.

Use a request similar to:

```
GET https://vcloud.example.com/api/admin/extension
```

The response contains a `Link` element for retrieving NSX Manager instances and operations similar to:

```
<Link
   type="application/vnd.vmware.admin.networkManagers+xml"
   rel="down"
   href="https://vcloud.example.com/api/admin/extension/nsxtManagers"/>
```

2 Retrieve the XML representation of the NSX Manager instances and operations by using the URL from the response in Step 1.

Use a request similar to:

```
GET https://vcloud.example.com/api/admin/extension/nsxtManagers
Content-Type: application/vnd.vmware.admin.networkManagers+xml
...
```

The response contains a `Link` element for adding NSX Manager instances similar to:

```
<Link
   type="application/vnd.vmware.admin.nsxTmanager+xml"
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/nsxtManagers"/>
```

3 Create an `NsxTManager` element that includes the information required to register the NSX Manager instance.

4 POST the `NsxTManager` element that you created in Step 3 to the URL from the response in Step 2.

See the request portion of Register an NSX Manager Instance.

Example: Register an NSX Manager Instance

You must supply the user name and password of the NSX Manager administrator in the request. The response includes NSX URLs for the newly registered NSX Manager instance, and omits the password.

Request:

```
POST https://vcloud.example.com/api/admin/extension/nsxtManagers
Content-Type: application/vnd.vmware.admin.nsxTmanager+xml
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:NsxTManager
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ns2="http://www.vmware.com/vcloud/extension/v1.5"
   name="nsxManager1">
   <Description>NSX-T Manager</Description>
   <ns2:Username>admin</ns2:Username>
   <ns2:Password>Welcome@123</ns2:Password>
   <ns2:Url>https://10.100.121.67</ns2:Url>
</ns2:NsxTManager>
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.networkmanagers+xml;version=31.0
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vmext:NsxTManager
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   name="nsxManager1"
   id="urn:vcloud:nsxtmanager:24"
   type="application/vnd.vmware.admin.nsxTmanager+xml"
   <Link
      rel="edit"
      href="http://www.vmware.com/api/admin/extension/nsxtManagers/24
      type="application/vnd.vmware.admin.nsxTmanager+xml"/>
   <Link
      rel="remove"
      href="http://www.vmware.com/api/admin/extension/nsxtManagers/24
      type="application/vnd.vmware.admin.nsxTmanager+xml"/>
   <Link
      rel="up"
      href="https://wdc-vcd-sp-static-34-90.eng.vmware.com/api/admin/extension/nsxtManagers"
      type="application/vnd.vmware.admin.networkManagers+xml"/>
   <Description>NSX-T Manager</Description>
   <vmext:Username>admin</vmext:Username>
   <vmext:Url>https://10.100.121.67</vmext:Url>
</vmext:NsxTManager>
```

What to do next

Create a provider VDC backed by the registered NSX instance.

Finding Available vCenter Resources

Many of the operations required to import virtual machines or create Provider VDCs, external networks, and network pools require you to identify vCenter resources and obtain references to them. You use these references to make the vCenter resources available in the cloud.

Every vCenter server registered to your cloud is represented as a `VimServerReference` element in the cloud's `vimServerReferences` list. You can retrieve one of these references to get a detailed representation of the server object, including links to the server's resource pools, networks, hosts, and virtual machines.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the list of vCenter servers registered to this cloud.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension/vimServerReferences
```

2 Retrieve the representation of a vCenter server.

The response to the request you made in Step 1 contains a list of `VimServerReference` elements. You can make a GET request to any of these references to retrieve the XML representation of a vCenter server registered to this cloud.

Results

The `VimServer` element returned in response to the request you made in Step 2 includes several `Link` elements where `rel="down"`. These links contain URLs that you can use to retrieve lists of references to vCenter resources on this server.

Table 8-1. vCenter Resource Lists

List URL	List Contents
`https://vcloud.example.com/api/admin/extension/vimServer/`*`id`*`/vmsList`	References to virtual machines in this vCenter server's inventory
`https://vcloud.example.com/api/admin/extension/vimServer/`*`id`*`/hostReferences`	References to hosts in this vCenter server's inventory
`https://vcloud.example.com/api/admin/extension/vimServer/`*`id`*`/resourcePoolList`	`ResourcePool` objects
`https://vcloud.example.com/api/admin/extension/vimServer/`*`id`*`/networks`	`VimObjectRef` elements of type `DV_PORTGROUP` and `NETWORK`
`https://vcloud.example.com/api/admin/extension/vimServer/`*`id`*`/storageProfiles`	`VMWStorageProfile` objects.

The `VimServer` element returned in response also contains the `providerScoped` and `tenantScoped` elements, which provide information about the use of the vCenter Server resources.

- If the vCenter Server instance backs a provider VDC, its `providerScoped` element contains the value `true`.
- If the vCenter Server instance backs an SDDC, its `tenantScoped` element contains the value `true`.

Before you begin using the resources of an attached vCenter Server instance in the cloud, both elements, the `providerScoped` and the `tenantScoped`, contain the value `false`.

Example: Resources on a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9
```

Response:

```
200 OK
...
<vmext:VimServer
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="VC22"
   id="urn:vcloud:vimserver:9"
   type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
   href="https://vcloud.example.com/api/admin/extension/vimServer/9" ...>
   ...
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmsObjectRefsList+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/vmsList" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwHostReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/hostReferences" />
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.resourcePoolList+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/resourcePoolList" />
    <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vimServerNetworks+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/networks" />
    <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwStorageProfiles+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/storageProfiles" />
   ...
   <Description>vCenter Server</Description>
   <vmext:Username>administrator</vmext:Username>
   <vmext:Url>https://10.115.124.37:443</vmext:Url>
```

```
    <vmext:IsEnabled>true</vmext:IsEnabled>
    <vmext:IsConnected>true</vmext:IsConnected>
    <vmext:ShieldManagerHost>10.115.124.1</vmext:ShieldManagerHost>
    <vmext:ShieldManagerUserName>admin</vmext:ShieldManagerUserName>
    <vmext:Uuid>44D5DAAA-7F3E-456D-B1CB-8288D7308AD6</vmext:Uuid>
    <vmext:VcProxy>cell1</vmext:VcProxy>
    <vmext:VcVersion>6.0.0</vmext:VcVersion>
    <vmext:UseVsphereService>false</vmext:UseVsphereService>
    <vmext:VsphereWebClientServerUrl>https://10.115.124.37</vmext:VsphereWebClientServerUrl>
</vmext:VimServer>
```

Retrieve a List of Resource Pools from a vCenter Server

You can retrieve the list of resource pools available on a vCenter server registered to a cloud. To retrieve the list, you make a GET request to the server's `resourcePoolList` link.

The `ResourcePoolList` of a `VimServer` element contains an entry for every available resource pool on the server. Resource pools that a provider VDC is already using are not listed, because they are considered unavailable. See Finding Available vCenter Resources.

Prerequisites

- This operation is restricted to system administrators.
- Retrieve the XML representation of a vCenter server registered to your cloud. See Finding Available vCenter Resources.

Procedure

1. Examine the `VimServer` element to locate its `resourcePoolList` link.

   The link has the following form:

   ```
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.resourcePoolList+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/resourcePoolList" />
   ```

2. GET the URL in the value of this link's `href` attribute to retrieve the list of resource pools.

   See Retrieve a List of Resource Pools from a vCenter Server. If the list is empty, all resource pools on the server are already in use.

Example: Retrieve a List of Resource Pools from a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9/resourcePoolList
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.resourcepoollist+xml
...
<vmext:ResourcePoolList
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.resourcePoolList+xml" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
   <vmext:ResourcePool
      name="cluster2">
      <vmext:MoRef>resgroup-195</vmext:MoRef>
      <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      <vmext:DataStoreRefs>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name="vc9-ds1"
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-172</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name="vc9-ds2"
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-173</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
       </vmext:DataStoreRefs>
   </vmext:ResourcePool>
   <vmext:ResourcePool
      name="cluster3">
       <vmext:MoRef>resgroup-230</vmext:MoRef>
      <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      <vmext:DataStoreRefs>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name="vc9-ds10"
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-174</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
      </vmext:DataStoreRefs>
   </vmext:ResourcePool>
   ...
</vmext:ResourcePoolList>
```

Retrieve a List of Available Portgroups and Switches from a vCenter Server

To retrieve the list of available portgroups and switches from a vCenter server registered to VMware Cloud Director, make a GET request to the server's `networks` link.

Retrieving the `networks` link from a `VimServer` element returns a `VimObjectRefList` element that contains references to available `DV_SWITCH`, `DV_PORTGOUP`, and `NETWORK` objects on the server. Objects that have already been consumed in the creation of an external network are not listed. See Finding Available vCenter Resources.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of a vCenter server registered to VMware Cloud Director.

2 Examine the response, a `VimServer` element, to locate the `networks` link.

  This link has the following form:

```
<vcloud:Link
   rel="down"
   type="application/vnd.vmware.admin.vimServerNetworks+xml"
   href="https://vcloud.example.com/api/admin/extension/vimServer/9/networks" />
```

3 GET the URL in the value of this link's `href` attribute to retrieve the list of network resources.

  See Retrieve a List of Available Portgroups and Switches from a vCenter Server. If the list is empty, all network resources on the server are already in use.

Example: Retrieve a List of Available Portgroups and Switches from a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9/networks
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vimservernetworks+xml
...
<vmext:VimObjectRefList
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
   <vmext:VimObjectRefs>
      <vmext:VimObjectRef>
```

```
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="vc9"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>dvportgroup-32</vmext:MoRef>
         <vmext:VimObjectType>DV_PORTGROUP</vmext:VimObjectType>
      </vmext:VimObjectRef>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="vc9"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>dvportgroup-175</vmext:MoRef>
         <vmext:VimObjectType>DV_PORTGROUP</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:VimObjectRefs>
</vmext:VimObjectRefList>
```

Retrieve a List of External Networks and Network Pools

You can retrieve a list of external networks and network pools that have been created on a vCenter server registered to a cloud.

A reference to an external network is required when you create an Edge Gateway. A reference to a network pool is usually required when you create an organization VDC. These resources do not exist in a new VMware Cloud Director installation. A system administrator must create them, as described in Create an External Network and Create a Network Pool.

Note Every Provider VDC requires a `VxlanPoolType` network pool. You can choose to have the system create this pool automatically with a standard configuration, or you can create one yourself that has a custom configuration and have the system attach it to the new Provide VDC. See Create a Network Pool.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of your cloud.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/admin/extension
   ```

2 Examine the response, a `VMWExtension` element, to locate the links to lists of external networks and network pools.

   These links have the following form:

   ```
   <vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.vmwExternalNetworkReferences+xml"
   ```

```
   href="https://vcloud.example.com/api/admin/extension/externalNetworkReferences" />
<vcloud:Link
   rel="down"
   type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
   href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
```

3. Make a GET request to the link that represents the object type of interest.

   See Retrieve a List of External Networks.

Example: Retrieve a List of External Networks

Request:

```
GET https://vcloud.example.com/api/admin/extension/externalNetworkReferences
```

Each reference to an external network includes its `type`, `name`, and `href` attributes, as shown in this example.

Response:

```
<vmext:VMWExternalNetworkReferences ... >
   ...
   <vmext:ExternalNetworkReference
      type="application/vnd.vmware.admin.extension.network+xml"
      name="VC0"
      href="https://vcloud.example.com/api/admin/extension/externalnet/85" />
   <vmext:ExternalNetworkReference
      type="application/vnd.vmware.admin.extension.network+xml"
      name="VC1"
      href="https://vcloud.example.com/api/admin/extension/externalnet/302" />
   ...
</vmext:VMWExternalNetworkReferences>
```

The corresponding element for network pools, `VMWNetworkPoolReferences`, is similar. In most cases, you can supply just the `href` attribute value when you specify an external network or network pool in an organization network creation request. You can retrieve additional information about the external network or network pool by making a GET request to its `href` attribute value.

Retrieve a List of Virtual Machines from a vCenter Server

You can retrieve the list of virtual machines in the inventory of a vCenter server that is registered to a cloud. To retrieve the list, make a GET request to the server's `vmsList` link.

When you import a virtual machine from vCenter, your request must supply a reference to the vCenter server and a VIM object reference to the virtual machine. See Finding Available vCenter Resources.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of a vCenter server registered to your cloud.

2 Examine the response, a `VimServer` element, to locate the `vmsList` link.

This link has the following form:

```
<vcloud:Link
   rel="down"
   type="application/vnd.vmware.admin.vmsObjectRefsList+xml"
   href="https://vcloud.example.com/api/admin/extension/vimServer/9/vmsList" />
```

3 GET the URL in the value of this link's `href` attribute to retrieve the list of virtual machines.

See the request portion of Retrieve a List of Virtual Machines from a vCenter Server. If the list is empty, there are no virtual machines in the server's inventory.

Example: Retrieve a List of Virtual Machines from a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9/vmsList
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vmsobjectrefslist+xml
...
<vmext:VmObjectRefsList
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   page="1"
   numberOfPages="1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <vmext:VmObjectRef
      name="RH5u3_32bit">
      <vmext:VimServerRef
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         name="vc2-v41u1"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>vm-41</vmext:MoRef>
      <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
   </vmext:VmObjectRef>
   <vmext:VmObjectRef
      name="W2K3 64 R2">
      <vmext:VimServerRef
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         name="vc2-v41u1"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>vm-43</vmext:MoRef>
      <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
   </vmext:VmObjectRef>
   <vmext:VmObjectRef
      name="Ubuntu91_32_vt4">
      <vmext:VimServerRef
```

```
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         name="vc2-v41u1"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>vm-44</vmext:MoRef>
      <vmext:VimObjectType>VIRTUAL_MACHINE</vmext:VimObjectType>
   </vmext:VmObjectRef>
</vmext:VmObjectRefsList>
```

Retrieve a List of Storage Profiles from a vCenter Server

You can retrieve the list of storage profiles that have been defined on a vCenter server registered to a cloud. To retrieve the list, you make a GET request to the server's `storageProfiles` link.

Storage profiles are named configurations of vCenter storage. When you create a Provider VDC, you must specify the name of at least one storage profile to provide storage capacity for that Provider VDC.

The `VMWStorageProfiles` of a `VimServer` element contains an entry for every storage profile that has been defined on the server. See Finding Available vCenter Resources.

Note Storage profiles are represented as Storage Policies in the VMware Cloud Director HTML5 UI.

Prerequisites

- This operation is restricted to system administrators.
- Retrieve the XML representation of a vCenter server registered to your cloud. See Finding Available vCenter Resources.

Procedure

1 (Optional) Refresh the cached list of storage profiles.

VMware Cloud Director maintains a cache of the storage profiles that have been created on each of its registered vCenter servers. The cache is refreshed on a regular schedule, and is likely to contain an up-to-date list. If a vCenter administrator has recently created a new storage profile, you can force it to be added to the cache by following these steps.

a Examine the `VimServer` element to locate its `action/refreshStorageProfiles` link.

The link has the following form:

```
<vcloud:Link
   rel="refreshStorageProfiles"
   href="https://vcloud.example.com/api/admin/extension/vimServer/9/action/
refreshStorageProfiles" />
```

b Refresh the cached list of storage profiles from the vCenter server.

Use a request like this one.

```
POST https://vcloud.example.com/api/admin/extension/vimServer/9/action/
refreshStorageProfiles
```

The response is a `Task`. When the task completes, the cached list of storage profiles on the vCenter server has been updated.

2 Examine the `VimServer` element to locate its `storageProfiles` link.

The link has the following form:

```
<vcloud:Link
      rel="down"
      type="application/vnd.vmware.admin.storageProfiles+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/storageProfiles" />
```

3 GET the URL in the value of this link's `href` attribute to retrieve the list of storage profiles.

See Retrieve a List of Storage Profiles from a vCenter Server.

Example: Retrieve a List of Storage Profiles from a vCenter Server

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9/storageProfiles
```

This response shows that the specified vCenter server contains three storage profiles, named `Gold`, `Silver`, and `Bronze`. Storage profile details, including datastore identifiers and capacities, are included in the response, but only the value of the `name` attribute of a `VMWStorageProfile` is needed when adding that storage profile to a Provider VDC.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vmwstorageprofiles+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWStorageProfiles
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   ...>
   ...
   <vmext:VMWStorageProfile
      name="Gold">
      <vmext:MoRef>storageProfile-CFAA6D92-36FC-4C16-9B30-FAC79B902371</vmext:MoRef>
      <vmext:VimObjectType>STORAGE_PROFILE</vmext:VimObjectType>
      <vmext:FreeStorageMb>169203.0</vmext:FreeStorageMb>
      <vmext:TotalStorageMb>325120.0</vmext:TotalStorageMb>
      <vmext:DataStoreRefs>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name=""
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-44</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
         <vmext:VimObjectRef>
            <vmext:VimServerRef
               type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
               name=""
               href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
            <vmext:MoRef>datastore-45</vmext:MoRef>
            <vmext:VimObjectType>DATASTORE</vmext:VimObjectType>
         </vmext:VimObjectRef>
      </vmext:DataStoreRefs>
   </vmext:VMWStorageProfile>
   <vmext:VMWStorageProfile
      name="Silver">
      ...
   </vmext:VMWStorageProfile>
   <vmext:VMWStorageProfile
      name="Bronze">
      ...
   </vmext:VMWStorageProfile>
</vmext:VMWStorageProfiles>
```

Create an Organization

To create an organization, a system administrator POSTs an `AdminOrg` element to the cloud's `add` URL for `orgs`.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the cloud.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin
```

2 Examine the response to locate the `Link` element that contains the URL for adding organizations to the cloud.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.organization+xml`, as shown here:

```
<Link href="https://vcloud.example.com/api/admin/orgs"
   rel="add"
   type="application/vnd.vmware.admin.organization+xml"/>
```

3 Create an `AdminOrg` element that specifies the properties of the organization.

See the request portion of Create an Organization.

4 POST the `AdminOrg` element you created in Step 3 to the URL described in Step 2.

See the request portion of Create an Organization.

Results

The server creates and enables the organization, and returns an `AdminOrg` element that includes the contents you POSTed, along with a set of `Link` elements that you can use to access, remove, disable, or modify it. VMware Cloud Director API users can log in to this organization using the URL specified in the `href` attribute of the `Link` where `rel="alternate"`. Users of the VMware Cloud Director Tenant Portal can log in to the organization at a URL of the form *cloud-url*/org/*name*, where *cloud-url* is a URL of the form `https://vcloud.example.com/cloud` and *name* is the value of the `name` attribute of the `AdminOrg` element. To log in to the organization created by Create an Organization, a user opens a browser and navigates to `https://vcloud.example.com/cloud/org/Finance`.

Example: Create an Organization

This request creates an organization and specifies a few of its properties. Although this example populates the `Settings` element with a few typical values, the request would be valid even if you left `Settings` empty. For a list of all required and optional elements that an `AdminOrg` contains, see the schema reference.

Request:

```
POST https://vcloud.example.com/api/admin/orgs
Content-Type: application/vnd.vmware.admin.organization+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminOrg
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Finance"
   type="application/vnd.vmware.admin.organization+xml">
   <Description>Example Corporation’s Finance Organization</Description>
   <FullName>Finance</FullName>
   <Settings>
      <OrgGeneralSettings>
         <CanPublishCatalogs>false</CanPublishCatalogs>
         <CanPublishExternally>true</CanPublishExternally>
          <CanSubscribe>false</CanSubscribe>
         <DeployedVMQuota>0</DeployedVMQuota>
         <StoredVmQuota>0</StoredVmQuota>
         <UseServerBootSequence>false</UseServerBootSequence>
         <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
      </OrgGeneralSettings>
      <OrgLdapSettings>
         <OrgLdapMode>SYSTEM</OrgLdapMode>
         <CustomUsersOu />
      </OrgLdapSettings>
      <OrgEmailSettings>
         <IsDefaultSmtpServer>true</IsDefaultSmtpServer>
         <IsDefaultOrgEmail>true</IsDefaultOrgEmail>
         <FromEmailAddress />
         <DefaultSubjectPrefix />
         <IsAlertEmailToAllAdmins>true</IsAlertEmailToAllAdmins>
        </OrgEmailSettings>
   </Settings>
</AdminOrg>
```

The response contains information extracted from the request, and includes links that an administrator can use to manage the organization and its settings, and to add resources such as VDCs, catalogs, and users. On creation, `AdminOrg` objects are disabled by default unless the create request includes an `IsEnabled` element with a value of `true`. A system administrator must enable a disabled `AdminOrg` before users can log into it.

The response also includes elements inherited from system defaults, including:

- `OrgPasswordPolicySettings`
- `VAppLeaseSettings`
- `VAppTemplateLeaseSettings`
- `RightReferences`
- `RoleReferences`

.

Note When you create an organization, it includes `RoleReferences` for all predefined roles and a set of `RightReferences` that contains all rights included in those roles. You can include an explicit set of `RightReferences` in the `AdminOrg` request body if you want the organization to include a set of rights that differs from the set it would otherwise receive on creation. If you do this, the predefined roles are created in the organization with a set of rights that is the intersection of the rights in the predefined role and the rights you granted to the organization. A system administrator can add or remove rights after an organization has been created. See Edit Organization Rights.

This example shows only a subset of the actual response.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.organization+xml
...
<AdminOrg
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Finance"
   id="urn:vcloud:org:26"
   type="application/vnd.vmware.admin.organization+xml"
   href="https://vcloud.example.com/api/admin/org/26">
   <Link
      rel="add"
      href="https://vcloud.example.com/api/admin/org/26/catalogs"
      type="application/vnd.vmware.admin.catalog+xml" />
   <Link
      rel="add"
      href="https://vcloud.example.com/api/admin/org/26/users"
      type="application/vnd.vmware.admin.user+xml" />
   ...
   <Link
      rel="add"
      href="https://vcloud.example.com/api/admin/org/26/roles"
      type="application/vnd.vmware.admin.role+xml" />
   <Description>Example Corporation's Finance Organization</Description>
   <FullName>Finance</FullName>
   <IsEnabled>false</IsEnabled>
   <Settings ...>
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/vAppTemplateLeaseSettings"
         type="application/vnd.vmware.admin.vAppTemplateLeaseSettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/email"
         type="application/vnd.vmware.admin.organizationEmailSettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/vAppLeaseSettings"
         type="application/vnd.vmware.admin.vAppLeaseSettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/passwordPolicy"
         type="application/vnd.vmware.admin.organizationPasswordPolicySettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/general"
         type="application/vnd.vmware.admin.organizationGeneralSettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/ldap"
         type="application/vnd.vmware.admin.organizationLdapSettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/
guestPersonalizationSettings"
         type="application/vnd.vmware.admin.guestPersonalizationSettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/operationLimitsSettings"
         type="application/vnd.vmware.admin.operationLimitsSettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/federation"
         type="application/vnd.vmware.admin.organizationFederationSettings+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/org/26/settings/oauth"
         type="application/vnd.vmware.admin.organizationOAuthSettings+xml" />
      <Link
         rel="edit"
         href="https://vcloud.example.com/api/admin/org/26/settings"
         type="application/vnd.vmware.admin.orgSettings+xml" />
      ...
      <OrgGeneralSettings>
         <CanPublishCatalogs>false</CanPublishCatalogs>
         <CanPublishExternally>true</CanPublishExternally>
          <CanSubscribe>false</CanSubscribe>
         <DeployedVMQuota>0</DeployedVMQuota>
         <StoredVmQuota>0</StoredVmQuota>
         <UseServerBootSequence>false</UseServerBootSequence>
         <DelayAfterPowerOnSeconds>0</DelayAfterPowerOnSeconds>
         ...
      </OrgGeneralSettings>
         ...
      <OrgLdapSettings>
         <OrgLdapMode>SYSTEM</OrgLdapMode>
         <CustomUsersOu />
      </OrgLdapSettings>
      <OrgEmailSettings>
         <IsDefaultSmtpServer>true</IsDefaultSmtpServer>
         <IsDefaultOrgEmail>true</IsDefaultOrgEmail>
         <FromEmailAddress />
         <DefaultSubjectPrefix />
         <IsAlertEmailToAllAdmins>true</IsAlertEmailToAllAdmins>
         ...
        </OrgEmailSettings>
         ...
      </Settings>
      <RightReferences ...>
         ...
      </RightReferences>
      <RoleReferences>
         ...
      </RoleReferences>
</AdminOrg>
```

Edit Organization Rights

An organization is initially granted all rights contained in the predefined roles. A system administrator can grant additional rights to an organization or remove rights previously granted.

Many newer VMware Cloud Director features require roles that use them to have special rights. These rights are not granted to an organization when you create it, or when you upgrade to a new VMware Cloud Director release. A system administrator must grant these rights to organizations that are entitled to use the features, and an organization administrator must add the rights to existing roles or create new roles that include them.

There are several ways to edit the rights in an organization:

- You can add rights to the organization. This operation updates the set of rights already in the organization by adding new ones. No rights are removed.
- You can replace the existing set of rights with a new set of rights. The replacement set of rights typically combines the rights that currently exist in the organization with additional rights available from the system.
- You can remove individual rights or groups of rights from the organization.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the set of rights available in the system.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin
```

The `RightReferences` element returned by this request includes a `RightReference` for each right that you can grant to an organization.

2 Retrieve the set of rights that exist in the organization.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/id/rights
```

The `OrgRights` element returned by this request includes a `RightReference` for each right that exists in the organization.

3 Update the organization's rights.

Option	Description
Add new rights.	Create an `OrgRights` request body that the new rights (a subset of the rights returned by request you made in Step 1) that you want to grant to the organization. POST the modified `OrgRights` request body to the organization's `edit` link for rights.
Replace all rights.	Create an `OrgRights` request body that includes the rights returned by the request you made in Step 2 and the additional rights (a subset of the rights returned by request you made in Step 1) that you want to grant to the organization. PUT the modified `OrgRights` request body to the organization's `edit` link for rights.
Remove a right with identifier *id*.	Make a request like ths one: `DELETE https://vcloud.example.com/api/admin/org/26/right/`*`id`*

Example: Grant Additional Rights to an Organization

This request adds several unassigned rights (rights that are not part of any predefined role) to an organization.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/rights
Content-type: application/vnd.vmware.admin.org.rights+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgRights xmlns="http://www.vmware.com/vcloud/v1.5">
   <RightReference
      href="https://vcloud.example.com/api/admin/right/79f48ce4-975d-3b88-ad19-a25ce58b9e91"
      name="vApp: Allow All Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/896288a1-6dc9-39d1-b2e8-89332a93f97d"
      name="vApp: Allow Ethernet Coalescing Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/a98dd24b-8fd0-3fac-b548-611a467a53df"
      name="vApp: Allow Latency Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/a96bebdd-ee34-3d62-9014-5283a88bad34"
      name="vApp: Allow Matching Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
</OrgRights>
```

This variant of the request adds the same set of rights to the organization, but replaces the rights in the organization with the set of rights in the request body. Any rights not present in the request body are removed from the organization, and from all organization roles that include them.

Request:

```
PUT https://vcloud.example.com/api/admin/org/26/rights
Content-type: application/vnd.vmware.admin.org.rights+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgRights xmlns="http://www.vmware.com/vcloud/v1.5">
   <!-- include the existing rights -->
   <RightReference
      href="https://vcloud.example.com/api/admin/right/5ddb661d-caf0-3680-9a74-59d4b06137f3"
      name="Disk: Change Owner"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/b755b050-772e-3c9c-9197-111c286f563d"
      name="Organization vDC Gateway: Configure Firewall"
      type="application/vnd.vmware.admin.right+xml" />
   ...

   <!-- then add the new rights -->
   <RightReference
      href="https://vcloud.example.com/api/admin/right/79f48ce4-975d-3b88-ad19-a25ce58b9e91"
      name="vApp: Allow All Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/896288a1-6dc9-39d1-b2e8-89332a93f97d"
      name="vApp: Allow Ethernet Coalescing Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/a98dd24b-8fd0-3fac-b548-611a467a53df"
      name="vApp: Allow Latency Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/a96bebdd-ee34-3d62-9014-5283a88bad34"
      name="vApp: Allow Matching Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
</OrgRights>
```

The response (only a portion of which is shown here) includes the new set of `OrgRights`. It also includes a link you can use to edit this list of rights.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgRights
   xmlns="http://www.vmware.com/vcloud/v1.5"
   href="https://vcloud.example.com/api/admin/org/798e24f6-b0bd-4b93-8e2e-c3c65c907f43/
rights">
   <Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/org/798e24f6-b0bd-4b93-8e2e-c3c65c907f43/
rights"
      type="application/vnd.vmware.admin.org.rights+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/5ddb661d-caf0-3680-9a74-59d4b06137f3"
      name="Disk: Change Owner"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/b755b050-772e-3c9c-9197-111c286f563d"
      name="Organization vDC Gateway: Configure Firewall"
      type="application/vnd.vmware.admin.right+xml" />
   ...
   <RightReference
      href="https://vcloud.example.com/api/admin/right/79f48ce4-975d-3b88-ad19-a25ce58b9e91"
      name="vApp: Allow All Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/896288a1-6dc9-39d1-b2e8-89332a93f97d"
      name="vApp: Allow Ethernet Coalescing Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/a98dd24b-8fd0-3fac-b548-611a467a53df"
      name="vApp: Allow Latency Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
   <RightReference
      href="https://vcloud.example.com/api/admin/right/a96bebdd-ee34-3d62-9014-5283a88bad34"
      name="vApp: Allow Matching Extra Config"
      type="application/vnd.vmware.admin.right+xml" />
</OrgRights>
```

Enable, Disable, or Remove an Organization

An `AdminOrg` element includes `action` links that a system administrator can use to enable, disable, or remove the organization.

Prerequisites

- This operation is restricted to system administrators.
- Retrieve the XML representation of the organization. See Retrieve a List of Organizations Accessible to You.

Procedure

- To enable an organization, POST a request to its `action/enable` link.
- To disable an organization, POST a request to its `action/disable` link.

- To remove an organization, remove all the objects it contains, then disable and remove it.

   a Delete all of the following from the organization:

   - local users and groups
   - catalogs
   - VDCs

   b POST a request to the `action/disable` link to disable the organization.

   After the organization is disabled, its representation includes a `rel="remove"` link.

   c Make a DELETE request to the organization's `rel="remove"` link.

Note You can make a request like this one, which adds the query string `recursive=true` to the `Org` href, to remove an organization that contains one or more objects as long as those objects are in a state that normally allows removal.

```
DELETE https://vcloud.example.com/api/admin/org/26?recursive=true
```

You can use an additional query parameter to force this kind of recursive removal even when the organization contains objects that are not in an appropriate state for removal.

```
DELETE https://vcloud.example.com/api/admin/vdc/26?recursive=true&force=true
```

Results

The server takes the requested action and returns a `Task`.

Create a Provider VDC

A Provider VDC is a collection of compute, memory, and storage resources from one vCenter Server instance. For network resources, a Provider VDC can use NSX Data Center for vSphere or NSX. A Provider VDC provides resources to organization VDCs.

A Provider VDC is represented as a `VMWProviderVdc` element in the extension view and a `ProviderVdc` element in the admin view. A system administrator can create a `VMWProviderVdc` or modify it to add or remove datastores, storage profiles, and resource pools, or change other properties such as its description. A system administrator cannot change the primary resource pool or vCenter Server instance that was specified when the Provider VDC was created. An organization administrator can retrieve a read-only representation of a Provider VDC in a `ProviderVdc` element. The `ProviderVdc` element includes a subset of the elements and attributes that are visible in the corresponding `VMWProviderVdc`.

Prerequisites

- This operation is restricted to system administrators.

- Decide which vCenter Server instance to supply a resource pool and storage profiles to this Provider VDC. See Finding Available vCenter Resources.
- If you want the Provider VDC to use NSX Data Center for vSphere for network services, attach the vCenter Server instance together with the associated NSX Manager instance to your cloud. See #unique_184.
- If you want the Provider VDC to use NSX Data Center for vSphere with a custom VXLAN-backed network pool instead of the default VXLAN-backed network pool, Create a VXLAN-Backed Network Pool.
- If you want the Provider VDC to use NSX for network services, Register an NSX Manager Instance.

Procedure

1 Retrieve the XML representation of the extension objects and operations.

Use a request similar to:

```
GET https://vcloud.example.com/api/admin/extension
```

2 In the response, locate the `Link` element that contains the URL for adding Provider VDCs to the cloud.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.createProviderVdcParams+xml`, as shown here:

```
<Link
   type="application/vnd.vmware.admin.createProviderVdcParams+xml"
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/providervdcsparams"/>
```

3 Create a `VMWProviderVdcParams` element that specifies the properties of the Provider VDC.

a Include a `VimServer` element that references the vCenter Server instance to supply a resource pool and storage profiles to this Provider VDC.

b Include a `ResourcePoolRefs` element that specifies one resource pool.

The `ResourcePoolRef` must contain the `href` attribute value of the `VimServer` element that you created in 3.a, and the `MoRef` and `VimObjectType` values of the resource pool as they appear in the `ResourcePool` element from the resource pool list. See the request portion of Create a Provider VDC Backed by NSX Data Center for vSphere or NSX.

Note You must specify exactly one resource pool when you create the Provider VDC, which is the primary resource pool. After creating the Provider VDC, you can add more resource pools.

c Include at least one `StorageProfile` element that contains the name of a storage profile that has been defined on the vCenter server referenced in the `VimServer` element you created in 3.a.

d If you created a VXLAN-backed network pool that you want this Provider VDC to use instead of the default VXLAN-backed network pool, include a VxlanNetworkPool element that references the pool. See Create a VXLAN-Backed Network Pool.

e If you registered an NSX Manager instance that you want to this Provider VDC to use, include an NsxTManagerReference element.

4 POST the VMWProviderVdc element you created in Step 3 to the URL described in Step 2.

See the request portion of Create a Provider VDC Backed by NSX Data Center for vSphere or NSX.

Results

The server creates and enables the Provider VDC and returns a VMWProviderVdc element that includes information derived from the contents you POSTed, along with a set of Link elements that you can use to access, remove, disable, or modify the Provider VDC.

- The new Provider VDC becomes a member of the ProviderVdcReferences element of the VCloud.
- The resource pool you selected is removed from the resource pool list of the vCenter Server instance. You cannot include this resource pool in any other Provider VDC.
- Each storage profile you specified becomes the basis for a ProviderVdcStorageProfile object, and can be retrieved from the Provider VDC after it has been created, or by using a providerVdcStorageProfile query.
- If you did not include a include a VxlanNetworkPool element in the request body, the system creates a VxlanPoolType network pool on the vCenter server referenced by the VimServer element you created in 3.a and attaches it to the new Provider VDC.

Example: Create a Provider VDC Backed by NSX Data Center for vSphere or NSX

These examples create Provider VDCs specifying a resource pool extracted from the response portion of Retrieve a List of Resource Pools from a vCenter Server and a storage profile extracted from Retrieve a List of Storage Profiles from a vCenter Server. The vCenter Server instance that provides the resources (the resource pool whose MoRef is resgroup-195 and a storage profile named Gold) is referenced in the VimServerRef and VimServer elements.

The responses include a Task that tracks the creation of the Provider VDC, and a set of Link elements that you can use to operate on or modify the Provider VDC. It also includes a HighestSupportedHardwareVersion value set to the latest virtual hardware version supported by the primary resource pool, and a list of HostReferences identifying the ESX hosts in that resource pool.

- Create a Provider VDC backed by NSX Data Center for vSphere

  Because the request does not include a VxlanNetworkPool element, the system creates the provider VDC with the default VXLAN network pool.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdcsparams
Content-Type: application/vnd.vmware.admin.createProviderVdcParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWProviderVdcParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="PVDC-Example">
   <vcloud:Description>Example Provider VDC</vcloud:Description>
   <vmext:ResourcePoolRefs>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-195</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:ResourcePoolRefs>
   <vmext:VimServer
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
   <vmext:StorageProfile>Gold</vmext:StorageProfile>
</vmext:VMWProviderVdcParams>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.vmwprovidervdc+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWProviderVdc
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   status="0"
   name="PVDC-Example"
   id="urn:vcloud:providervdc:35"
   type="application/vnd.vmware.admin.vmwprovidervdc+xml" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwProviderVdcReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/providerVdcReferences" />
   <vcloud:Link
      rel="alternate"
      type="application/vnd.vmware.admin.providervdc+xml"
      href="https://vcloud.example.com/api/admin/providervdc/35" />
   <vcloud:Link
      rel="update:resourcePools"
      type="application/vnd.vmware.admin.resourcePoolSetUpdateParams+xml"
      href="https://vcloud.example.com/api/admin/extension/providervdc/35/action/
updateResourcePools" />
   ...
   <vcloud:Description>Example Provider VDC</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         ...
         operation="Creating Provider Virtual Datacenter PVDC-Example(35)">
         ...
      </vcloud:Task>
   </vcloud:Tasks>
   <vcloud:ComputeCapacity>
      <vcloud:Cpu>
         <vcloud:Units>MHz</vcloud:Units>
         <vcloud:Allocation>0</vcloud:Allocation>
         <vcloud:Total>0</vcloud:Total>
         <vcloud:Used>0</vcloud:Used>
         <vcloud:Overhead>0</vcloud:Overhead>
      </vcloud:Cpu>
      <vcloud:Memory>
         <vcloud:Units>MB</vcloud:Units>
         <vcloud:Allocation>0</vcloud:Allocation>
         <vcloud:Total>0</vcloud:Total>
         <vcloud:Used>0</vcloud:Used>
         <vcloud:Overhead>0</vcloud:Overhead>
      </vcloud:Memory>
   </vcloud:ComputeCapacity>
   <AvailableNetworks>
      <Network
         type="application/vnd.vmware.admin.network+xml"
         name="VC1-VLAN48"
         href="https://vcloud.example.com/api/admin/network/297" />
   </AvailableNetworks>
   <StorageProfiles>
      <ProviderVdcStorageProfile
         type="application/vnd.vmware.admin.pvdcStorageProfile+xml"
         name="Gold"
         href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101" />
   </StorageProfiles>
   <IsEnabled>true</IsEnabled>
   <vcloud:NetworkPoolReferences>
      <vcloud:NetworkPoolReference
         type="application/vnd.vmware.admin.networkPool+xml"
         name="VXLAN01"
         href="https://vcloud.example.com/api/admin/extension/networkPool/
69c0a96e-1151-439d-b8f6-2e2a11785c9f" />
   </vcloud:NetworkPoolReferences>
   <vmext:DataStoreRefs />
   <vmext:ResourcePoolRefs>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="VC-A"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-195</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:ResourcePoolRefs>
   <vmext:VimServer
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
      name="ConfigWizard Configured vCenter"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
   <vmext:HostReferences>
      <vmext:HostReference
         type="application/vnd.vmware.admin.host+xml"
         name="ESX01.example.com"
         href="https://vcloud.example.com/api/admin/extension/host/83" />
      <vmext:HostReference
         type="application/vnd.vmware.admin.host+xml"
         name="ESX02.example.com"
         href="https://vcloud.example.com/api/admin/extension/host/261" />
   </vmext:HostReferences>
   <vmext:HighestSupportedHardwareVersion>vmx-07</vmext:HighestSupportedHardwareVersion>
</vmext:VMWProviderVdc>
```

- Create a Provider VDC backed by NSX

  Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdcsparams
Content-Type: application/vnd.vmware.admin.createProviderVdcParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ns2:VMWProviderVdcParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:ns2="http://www.vmware.com/vcloud/extension/v1.5"
   name="nsxTPvdc1">
   <ns2:ResourcePoolRefs>
      <ns2:VimObjectRef>
      <ns2:VimServerRef
         href="https://vcloud.example.com/api/admin/extension/vimServer/9"/>
         <ns2:MoRef>resgroup-195</ns2:MoRef>
         <ns2:VimObjectType>RESOURCE_POOL</ns2:VimObjectType>
      </ns2:VimObjectRef>
   </ns2:ResourcePoolRefs>
<ns2:VimServer
   href="https://vcloud.example.com/api/admin/extension/vimServer/9"
   id="urn:vcloud:vimserver:9"
   name="vc1"
   type="application/vnd.vmware.admin.vmwvirtualcenter+xml"/>
<ns2:NsxTManagerReference
   href="https://vcloud.example.com/api/admin/extension/nsxtManagers/24"
   id="urn:vcloud:nsxtmanager:24"
   name="nsxManager1"
   type="application/vnd.vmware.admin.nsxTmanager+xml"/>
<ns2:HighestSupportedHardwareVersion>vmx-7</ns2:HighestSupportedHardwareVersion>
<ns2:IsEnabled>true</ns2:IsEnabled>
<ns2:StorageProfile>*</ns2:StorageProfile>
</ns2:VMWProviderVdcParams>
```

  Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.vmwprovidervdc+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWProviderVdc
```

```
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_VirtualSystemSettingData"
   xmlns:common="http://schemas.dmtf.org/wbem/wscim/1/common"
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/
CIM_ResourceAllocationSettingData"
   xmlns:vmw="http://www.vmware.com/schema/ovf"
   xmlns:ovfenv="http://schemas.dmtf.org/ovf/environment/1"
   xmlns:ns9="http://www.vmware.com/vcloud/versions"
   status="0"
   name="nsxTPvdc1"
   id="urn:vcloud:providervdc:36"
   href="https://vcloud.example.com/api/admin/extension/providervdc/36"
   type="application/vnd.vmware.admin.vmwprovidervdc+xml">
   <Link
      rel="up"
      href="https://vcloud.example.com/api/admin/extension/providerVdcReferences"
      type="application/vnd.vmware.admin.vmwProviderVdcReferences+xml"/>
   <Link
      rel="edit"
href="https://vcloud.example.com/api/admin/extension/providervdc/36/storageProfiles"
type="application/vnd.vmware.admin.updateProviderVdcStorageProfiles+xml"/>
   <Link
      rel="alternate"
      href="https://vcloud.example.com/api/admin/providervdc/36"
      type="application/vnd.vmware.admin.providervdc+xml"/>
   <Link
      rel="update:resourcePools"
      href="https://vcloud.example.com/api/admin/extension/providervdc/36/action/
updateResourcePools"
      type="application/vnd.vmware.admin.resourcePoolSetUpdateParams+xml"/>
   <Link
      rel="down"
      href="https://vcloud.example.com/api/admin/extension/providervdc/36/resourcePools"
      type="application/vnd.vmware.admin.vmwProviderVdcResourcePoolSet+xml"/>
   <Link
      rel="down"
      href="https://vcloud.example.com/api/admin/extension/providervdc/36/
availableStorageProfiles"
      type="application/vnd.vmware.admin.vmwStorageProfiles+xml"/>
   <Link
      rel="down"
      href="https://vcloud.example.com/api/admin/extension/providervdc/36/
discoverResourcePools"
      type="application/vnd.vmware.admin.vmwDiscoveredResourcePools+xml"/>
   <Link
      rel="down"
      href="https://vcloud.example.com/api/admin/providervdc/36/vdcReferences"
      type="application/vnd.vmware.admin.vdcReferences+xml"/>
   <Link
      rel="down"
      href="https://vcloud.example.com/api/admin/providervdc/36/metadata"
      type="application/vnd.vmware.vcloud.metadata+xml"/>
```

```
    Link
      rel="down"
      href="https://vcloud.example.com/api/admin/providervdc/36/extension"
      type="application/vnd.vmware.admin.extensibility.selectors+xml"/>
   <Link
      rel="add"
      href="https://vcloud.example.com/cloudapi/providervdc/urn:vcloud:providervdc:36/
computePolicies"
      type="application/json"/>
   <Link
      rel="down"
      href="https://vcloud.example.com/cloudapi/providervdc/urn:vcloud:providervdc:36/
computePolicies"
      type="application/json"/>
   <Tasks>
      <Task
      ...
      operation="Creating Provider Virtual Datacenter nsxTPvdc1(36)"
      ...
      </Task>
   </Tasks>
   <ComputeCapacity>
      <Cpu>
         <Units>MHz</Units>
         <Total>0</Total>
      </Cpu>
      <Memory>
         <Units>MB</Units>
         <Total>0</Total>
      </Memory>
   </ComputeCapacity>
   <AvailableNetworks/>
   <StorageProfiles/>
   <IsEnabled>true</IsEnabled>
   <NetworkPoolReferences/>
   <vmext:DataStoreRefs/>
   <vmext:ResourcePoolRefs/>
   <vmext:VimServer
      href="https://vcloud.example.com/api/admin/extension/vimServer/9"
      id="9"
      name="vc1"
      type="application/vnd.vmware.admin.vmwvirtualcenter+xml"/>
   <vmext:NsxTManagerReference
      href="https://vcloud.example.com/api/admin/extension/nsxtManagers/24"
      id="24"
      name="nsxManager1"
      type="application/vnd.vmware.admin.nsxTmanager+xml"/>
   <vmext:HostReferences/>
   <vmext:HighestSupportedHardwareVersion>vmx-07</vmext:HighestSupportedHardwareVersion>
</vmext:VMWProviderVdc>
```

Retrieve a Provider VDC Resource Pool Set

The `VMWProviderVdcResourcePoolSet` of a Provider VDC contains information about all of the Provider VDC's resource pools. Getting this information is usually a prerequisite to adding or removing a resource pool.

Each reference in a `VMWProviderVdcResourcePoolSet` lists the vCenter server that provides the resource pool and indicates whether the resource pool is primary. All resource pools in a `VMWProviderVdcResourcePoolSet` must come from the same vCenter server.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the Provider VDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

2 Locate the `resourcePools` link in the `VMWProviderVdc`.

Every `VMWProviderVdc` element includes a link like this one to the Provider VDC's resource pools.

```
<Link
     rel="down"
     type="application/vnd.vmware.admin.vmwProviderVdcResourcePoolSet+xml"
     href="https://vcloud.example.com/api/admin/extension/providervdc/35/resourcePools"/>
```

3 Retrieve the `VMWProviderVdcResourcePoolSet` for the Provider VDC.

See Retrieve a Resource Pool Set.

Example: Retrieve a Resource Pool Set

This example lists the resource pools for the Provider VDC created in Create a Provider VDC Backed by NSX Data Center for vSphere or NSX. The response is a `VMWProviderVdcResourcePoolSet` that contains two resource pools, one of which is designated primary. Both reference the same vCenter server at `https://vcloud.example.com/api/admin/extension/vimServer/9`.

Request:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35/resourcePools
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.vmwprovidervdcresourcepoolset+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWProviderVdcResourcePoolSet
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwprovidervdc+xml"
      href="https://vcloud.example.com/api/admin/extension/providervdc/35" />
   <vmext:VMWProviderVdcResourcePool
      primary="true">
      <vcloud:Link
         rel="migrateVms"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/64/action/
migrateVms" />
      <vcloud:Link
         rel="resourcePoolVmList"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/64/vmList" />
      <vmext:ResourcePoolVimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="ConfigWizard Configured vCenter"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-235</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:ResourcePoolVimObjectRef>
      <vmext:ResourcePoolRef
         type="application/vnd.vmware.admin.vmwProviderVdcResourcePool+xml"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/64" />
      <vmext:Enabled>true</vmext:Enabled>
   </vmext:VMWProviderVdcResourcePool>
</vmext:VMWProviderVdcResourcePoolSet>
```

Update Provider VDC Resource Pools

A system administrator can update the resource pool set of an existing Provider VDC to add or remove resource pools. Adding resource pools allows organization VDCs that reference the Provider VDC to consume additional resources during periods of high demand. Removing resource pools frees the underlying resources.

When you create a Provider VDC, it initially contains a single resource pool, called the primary resource pool. Adding secondary resource pools allows a Provider VDC to support resource elasticity in organization VDCs that use the `AllocationPool` or `AllocationVApp` (pay as you go) allocation model. Resource elasticity allows an organization VDC's compute resources to grow or shrink on demand.

All of a Provider VDC's resource pools must come from the same vCenter. See Retrieve a List of Resource Pools from a vCenter Server.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the Provider VDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

2 Locate the `updateResourcePools` link in the `VMWProviderVdc`.

Every `VMWProviderVdc` element includes an action link like this one to the Provider VDC's `updateResourcePools` action.

```
<Link
     rel="update:resourcePools"
     type="application/vnd.vmware.admin.vmwProviderVdcResourcePoolSet+xml"
     href="https://vcloud.example.com/api/admin/extension/providervdc/35/action/
updateResourcePools"/>
```

3 Retrieve the resource pool list from the Provider VDC.

The `VMWProviderVdcResourcePoolSet` contains references to the Provider VDC's existing resource pools and the vCenter server that hosts them.

4 Update the resource pool set.

To add resource pools:

a Choose another resource pool from the same vCenter server.

b Create an `UpdateResourcePoolSetParams` element that contains an `AddItem` element for each resource pool to add.

To remove resource pools:

a Examine the resource pool list and find the pool to remove.

b Verify the pool is not the primary resource pool, and that no virtual machines are using it.

If necessary, use the `action/migrateVms` link to migrate virtual machines to another resource pool.

c Create an `UpdateResourcePoolSetParams` element that contains a `DeleteItem` element for each resource pool to remove.

5 POST the `UpdateResourcePoolSetParams` element to the Provider VDC's `resourcePools` link.

6 (Optional) Update the value of `HighestSupportedHardwareVersion` in the Provider VDC.

When you create a Provider VDC, the system sets its `HighestSupportedHardwareVersion` to the highest virtual hardware version supported by all hosts in the primary resource pool. When you add or remove resource pools, you might need to change this value to one that's valid for all hosts in the new resource pool set.

Example: Update Provider VDC Resource Pools

This request adds a resource pool to the Provider VDC created in Create a Provider VDC Backed by NSX Data Center for vSphere or NSX. The additional resource pool is hosted on the same vCenter server that hosts the existing resource pool. See Retrieve a List of Resource Pools from a vCenter Server for an example that lists the resource pools available on that server.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdc/35/action/updateResourcePools
Content-Type: application/vnd.vmware.admin.resourcePoolSetUpdateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:UpdateResourcePoolSetParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:AddItem>
      <vmext:VimServerRef
         type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>resgroup-230</vmext:MoRef>
      <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
   </vmext:AddItem>
</vmext:UpdateResourcePoolSetParams>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
   ...
</Task>
```

This request removes one of the two resource pools (a secondary resource pool) shown in Retrieve a Provider VDC Resource Pool Set. The response is a task.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdc/35/action/updateResourcePools
Content-Type: application/vnd.vmware.admin.resourcePoolSetUpdateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:UpdateResourcePoolSetParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:DeleteItem
      href="https://vcloud.example.com/api/admin/extension/resourcePool/66" />
</vmext:UpdateResourcePoolSetParams>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
...
</Task>
```

Update Provider VDC Storage Profiles

A system administrator can update the storage profiles that are included in a Provider VDC. New storage profiles can be added, and unused storage profiles can be removed.

Storage profiles can be created by a vCenter Server administrator on any vCenter Server instance that supports the profile-driven storage feature. A Provider VDC can provide access to any of the storage profiles that is created on its vCenter Server (the one referenced in its `vmext:VimServer` element). A VMware Cloud Director system administrator must specify at least one vCenter Server storage profile when creating a Provider VDC, and can add or remove storage profiles later as needed. Organization VDCs reference Provider VDC storage profiles in much the same way that Provider VDCs reference vCenter Server storage profiles. `Media` and `Disk` objects, as well as vApps and virtual machines reference organization VDC storage profiles by name.

Note Storage profiles are represented as Storage Policies in the VMware Cloud Director HTML5 UI.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 To retrieve the current storage profiles of the Provider VDC, use a request similar to:

  ```
  GET https://vcloud.example.com/api/admin/extension/providervdc/35
  ```

  The `ProviderVdc` response includes the `StorageProfiles` element, which contains a list of the storage profiles that are in use by the Provider VDC.

2 Create an `UpdateProviderVdcStorageProfiles` request body that specifies the details of the update.

  - To add a storage profile:

    a Retrieve the unused storage profiles that can be added to the Provider VDC.

      Use a request like this one:

      ```
      GET https://vcloud.example.com/api/admin/extension/providervdc/35/
      availableStorageProfiles
      ```

   b Create an `UpdateProviderVdcStorageProfiles` element that contains an `AddStorageProfile` element for each storage profile that you want to add.

     This request body adds a storage profile named `Bronze`:

```
<vmext:UpdateProviderVdcStorageProfiles
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:AddStorageProfile>Bronze</vmext:AddStorageProfile>
</vmext:UpdateProviderVdcStorageProfiles>
```

- To remove a storage profile:

   a Verify that no organization VDCs are using the storage profile you want to remove.

   b Disable the storage profile that you want to remove.

     Use a request like this one:

```
PUT https://vcloud.example.com/api/admin/extension/pvdcStorageProfile/87
<vmext:VMWProviderVdcStorageProfile
xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
...
type="application/vnd.vmware.admin.vmwPvdcStorageProfile+xml">
   <Enabled>false</Enabled>
   <Units>MB</Units>
</vmext:VMWProviderVdcStorageProfile>
```

   c Create an `UpdateProviderVdcStorageProfiles` element that contains a `RemoveStorageProfile` element for each storage profile to remove.

     This request body removes the storage profile with the UUID of *87*:

```
<vmext:UpdateProviderVdcStorageProfiles
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:RemoveStorageProfile
      href="https://vcloud.example.com/api/admin/extension/
pvdcStorageProfile/87" />
</vmext:UpdateProviderVdcStorageProfiles>
```

3 POST the `UpdateProviderVdcStorageProfiles` element to the Provider VDC's `storageProfiles` link.

  Use a request like this one:

```
POST https://vcloud.example.com/api/admin/extension/providervdc/35/storageProfiles
Content-Type: application/vnd.vmware.admin.updateProviderVdcStorageProfiles+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:UpdateProviderVdcStorageProfiles
...
</vmext:UpdateProviderVdcStorageProfiles>
```

Results

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
   ...
</Task>
```

The server returns a `Task` element that you can use to track the progress of the update. When the update is complete, the Provider VDC includes the updated set of storage profiles in its `StorageProfiles` element. Each storage profile you added becomes the basis for a `ProviderVdcStorageProfile` object, and can be retrieved from the Provider VDC after it has been created, or by using a `providerVdcStorageProfile` query.

Configure Storage I/O Control Support in a Provider VDC

If you want to enable specification of hard disk read/write performance by members of an organization, a Provider VDC that supports the organization must include a storage profile that is backed by an appropriately configured vSphere datastore.

Managed read/write performance in physical storage devices and virtual disks is defined in units called IOPS, which measure read/write operations per second. When an organization VDC storage profile is backed by a Provider VDC storage profile that includes storage devices that are capable of IOPS allocation, you can configure disks that use it to request a specified level of I/O performance. A storage profile configured with IOPS support delivers its default IOPS value to all disks that use it, even disks that are not configured to request a specific IOPS value. A hard disk configured to request a specific IOPS value cannot use a storage profile whose maximum IOPS value is lower than the requested value, or a storage profile that is not configured with IOPS support.

When backed by an appropriately configured Provider VDC storage profile, storage profiles in an organization VDC can be configured to support delivery of a specified level of I/O performance to disks that use them. See Configure Storage I/O Control Support in an Organization VDC.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Choose or create an appropriately configured vSphere storage policy.

Before VMware Cloud Director can enable IOPS for a Provider VDC storage profile, an IOPS-enabled vSphere storage policy must exist on a vCenter server registered to VMware Cloud Director.

- The storage devices backing the underlying vSphere datastores must be capable of IOPS support.

  Note You cannot enable IOPS support on a VMware Virtual SAN datastore.

- A vSphere administrator must configure the datastores with a specific vSphere custom field and value, as described in VMware Knowledge Base article http://kb.vmware.com/kb/2148300

- A vSphere administrator must create a vSphere storage policy that includes the IOPS-capable datastore.

2 Include the IOPS-capable vSphere storage profile in a Provider VDC.

Reference the IOPS-capable vSphere storage profile by name in a `ProviderVdcStorageProfile` element in the `VMWProviderVdcParams` request body you use when creating a Provider VDC or in the `UpdateProviderVdcStorageProfiles` element in an `updateStorageProfiles` request body you use when updating Provider VDC storage profiles.

Merge Provider VDCs

You can merge two Provider VDCs by specifying a target Provider VDC and a contributor Provider VDC to merge with it. The merged Provider VDC contains all resources from the target and contributor Provider VDCs

Because merging Provider VDCs is a resource-intensive operation, VMware Cloud Director allows only a single contributor to a merge. You can merge multiple Provider VDCs into a single target by making multiple merge requests, each specifying the same target and a new contributor.

When the merge is complete:

- Networks, network pools, storage profiles, resource pools, and datastores from the contributor Provider VDC are available in the target Provider VDC.
- Organization VDCs that were backed by the contributor are now backed by the target.
- The contributor Provider VDC is deleted.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Identify the Provider VDCs to merge.

Select a contributor that is not backed by the same resource pool as the target. You can use a query like this one to discover the resource pools backing each of your Provider VDCs.

```
GET https://vcloud.example.com/api/query?
type=providerVdcResourcePoolRelation&format=records
```

2 Construct a `ProviderVdcMergeParams` element whose `ProviderVdcReference` references the contributor.

See the request portion of Merge Provider VDCs.

3 Make a POST request to the `action/merge` link of the target Provider VDC and supply the `ProviderVdcMergeParams` as the request body.

Example: Merge Provider VDCs

This request merges the Provider VDC at `https://vcloud.example.com/api/admin/extension/providervdc/46` with the target Provider VDC specified in the request URL.

Request:

```
POST https://vcloud.example.com/api/admin/extension/providervdc/35/action/merge
Content-Type: application/vnd.vmware.admin.providerVdcMergeParams+xml
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vmext:ProviderVdcMergeParams
   type="application/vnd.vmware.admin.mergeParams+xml"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:ProviderVdcReference
      type="application/vnd.vmware.admin.vmwprovidervdc+xml"
      name="PVDC-VC001"
      href="https://vcloud.example.com/api/admin/extension/providervdc/46" />
</vmext:ProviderVdcMergeParams>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
   ...
</Task>
```

Create an External Network

An external network is a reference to a portgroup on a vCenter server attached to VMware Cloud Director. To create an external network, a system administrator must specify the vCenter server and a portgroup associated with it. External networks provide support for bridged organization networks.

Only a system administrator can create an external network. A system administrator can modify an external network to change properties such as its description, and to add or remove portgroups in the `VimPortGroupRefs` element. An organization administrator can retrieve a read-only representation of an external network to examine its properties.

Prerequisites

- This operation is restricted to system administrators.
- Retrieve the list of available portgroups. See Retrieve a List of Available Portgroups and Switches from a vCenter Server.

Procedure

1 Retrieve the XML representation of the vSphere platform extensions.

  Use a request like this one.

  ```
  GET https://vcloud.example.com/api/admin/extension
  ```

2 Examine the response to locate the `Link` element that contains the URL for adding external networks to the cloud.

  This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.vmwexternalnet+xml`, as shown here:

  ```
  <Link
     type="application/vnd.vmware.admin.vmwexternalnet+xml"
     rel="add"
     href="https://vcloud.example.com/api/admin/extension/externalnets"/>
  ```

3 Choose a vCenter server to provide a portgroup for the network.

4 Create a `VMWExternalNetwork` element that specifies the properties of the external network.

  These properties include the portgroup you specified in Step 3.

5 POST the `VMWExternalNetwork` element you created in Step 4 to the URL described in Step 2.

  See the request portion of Create an External Network.

Results

The server creates the external network and returns a `VMWExternalNetwork` element that includes the contents you POSTed, along with a set of `Link` elements that you can use to access, remove, disable, or modify it. A reference to the new external network is added to the `VMWExternalNetworkReferences` element of the `VCloud`. The portgroup you specified is removed from the `VimObjectRefList` of the vCenter server.

Example: Create an External Network

This request creates an external network backed by one of the portgroups listed in the response portion of Retrieve a List of Available Portgroups and Switches from a vCenter Server.

Request:

```
POST https://vcloud.example.com/api/admin/extension/externalnets
Content-Type: application/vnd.vmware.admin.vmwexternalnet+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWExternalNetwork
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="example-extnet"
   type="application/vnd.vmware.admin.vmwexternalnet+xml">
   <vcloud:Description>ExternalNet</vcloud:Description>
   <vcloud:Configuration>
      <vcloud:IpScopes>
         <vcloud:IpScope>
            <vcloud:IsInherited>false</vcloud:IsInherited>
            <vcloud:Gateway>10.24.64.126</vcloud:Gateway>
            <vcloud:Netmask>255.255.255.192</vcloud:Netmask>
            <vcloud:Dns1>10.115.120.71</vcloud:Dns1>
            <vcloud:Dns2>10.6.64.29</vcloud:Dns2>
            <vcloud:DnsSuffix>example.com</vcloud:DnsSuffix>
            <vcloud:IpRanges>
               <vcloud:IpRange>
               <vcloud:StartAddress>Start IP Address</vcloud:StartAddress>
               <vcloud:EndAddress>End IP Address</vcloud:EndAddress>
               </vcloud:IpRange>
            </vcloud:IpRanges>
         </vcloud:IpScope>
      </vcloud:IpScopes>
      <vcloud:FenceMode>isolated</vcloud:FenceMode>
   </vcloud:Configuration>
   <vmext:VimPortGroupRefs>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>dvportgroup-175</vmext:MoRef>
         <vmext:VimObjectType>DV_PORTGROUP</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:VimPortGroupRefs>
</vmext:VMWExternalNetwork>
```

You can specify more than one `VimObjectRef` in the `VimPortGroupRefs` as long as each `VimObjectRef` references a portgroup on a different vCenter Server registered to the system and all referenced objects have the same `VimObjectType` (`DV_PORTGROUP` or `NETWORK`).

The response includes a `Task` that tracks the creation of the network, and a set of `Link` elements that you can use to operate on or modify it.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin/vmwexternalnet+xml
...
<vmext:VMWExternalNetwork
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="example-extnet"
   id="urn:vcloud:network:85"
   type="application/vnd.vmware.admin.vmwexternalnet+xml"
   href="https://vcloud.example.com/api/admin/extension/externalnet/85"... >
   <vcloud:Link
      rel="alternate"
      type="application/vnd.vmware.admin.network+xml"
      href="https://vcloud.example.com/api/admin/network/85" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.vmwexternalnet+xml"
      href="https://vcloud.example.com/api/admin/extension/externalnet/85" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/externalnet/85" />
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwExternalNetworkReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/externalNetworkReferences" />
   <vcloud:Link
      rel="repair"
      type="application/vnd.vmware.admin.network+xml"
      href="https://vcloud.example.com/api/admin/network/85/action/reset" />
   <vcloud:Description>ExternalNet</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         startTime="2011-03-10T06:08:31.506-08:00"
         operationName="_network_create_provider_network"
         operation="Busy Network example-extnet(85)" ... >
         ...
       </vcloud:Task>
   </vcloud:Tasks>
   ...
</vmext:VMWExternalNetwork>
```

Create a Network Pool

Network pools provide support for isolated and routed networks in organization VDCs. Although every Provider VDC includes a VXLAN network pool that can support most networking use cases, a system administrator can create other types of network pools if they are needed.

A network pool object represents a collection of vSphere network resources that are contained by a Provider VDC and available to the organization VDCs backed by that Provider VDC. Traffic on each network in a pool is isolated at layer 2 from all other networks.

Only a system administrator can create a network pool. A system administrator can modify a network pool to change properties such as its description, but cannot change the network resources, such as virtual switches, portgroups, and NSX transport zones that provide backing for it. After a network pool has been associated with an organization VDC (typically when the VDC is created), network resources from the pool are consumed as needed to create isolated or routed organization VDC networks or vApp networks in the VDC.

All network pools are defined by a `VMWNetworkPool` element. The contents of this element depend on its type, which is specified in its `xsi:type` attribute. The following values of `xsi:type` are supported for pools created by a system administrator.

`VxlanPoolType`

This pool type is based on an NSX transport zone on a registered vCenter Server.

> **Note** Every Provider VDC requires a `VxlanPoolType` network pool. You can choose to have the system create this pool automatically with a standard configuration, or you can create one yourself that has a custom configuration and have the system attach it to the new Provide VDC.

`VlanPoolType`

This pool type is based on ESXi VLANs on a vCenter Server that backs the Provider VDC, and is backed by a range of VLAN IDs.

`PortGroupPoolType`

This pool type is based on distributed port groups of a vSphere distributed switch or third-party distributed switch on a vCenter Server that backs the Provider VDC.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the vSphere platform extensions.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension
```

2 Examine the `VMWExtension` response to locate the `Link` element that contains the URL for adding network pools to your cloud.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.networkPool+xml`, as shown here:

```
<Link
   type="application/vnd.vmware.admin.networkPool+xml"
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/networkPools"/>
```

3 Create a VMWNetworkPool element that specifies the pool type and backing vCenter Server resources.

   Details of this element's contents depend on the type of pool you are creating.

4 POST the VMWNetworkPool element you created in Step 3 to the URL described in Step 2.

Results

The server creates the network pool and returns a VMWNetworkPool element that includes the contents you POSTed, along with a set of Link elements that you can use to access, remove, disable, or modify it. A reference to the new network pool is added to the VMWNetworkPoolReferences element of the VCloud. Network resources you specified in the VMWNetworkPool element are removed from the VimObjectRefList of the vCenter Server.

Create a VXLAN-Backed Network Pool

To create a VXLAN-backed network pool, you create a VMWNetworkPool element whose type attribute has the value VXLANPoolType, and POST the element to your cloud's add link for networkPools.

A VXLAN network pool is required when creating a Provider VDC. By default, the system creates this network pool for you as part of the Provider VDC creation process. This default VXLAN network pool is scoped to a global NSX multicast transport zone, one that encompasses all clusters on the vCenter Server that backs the Provider VDC. If you want to create a Provider VDC whose VXLAN network pool has custom properties for NSX transport zone scope and control plane mode, create that network pool before you create the Provider VDC, then specify it in the VxlanNetworkPool element in the VMWProviderVdc request body. See Create a Provider VDC.

vSphere VXLAN networks are based on the IETF draft VXLAN standard. These networks support local-domain isolation equivalent to what is supported by vSphere isolation-backed networks. In addition, they provide:

- logical networks spanning layer 3 boundaries
- logical networks spanning multiple racks on a single layer 2
- broadcast containment
- higher performance
- greater scale (up to 16 million network addresses)

Prerequisites

- This operation is restricted to system administrators.
- Verify that you know your cloud's add URL for networkPools. See Create a Network Pool.
- Verify that at least one vCenter server attached to your cloud has network resources available. See Retrieve a List of Available Portgroups and Switches from a vCenter Server

Procedure

1 Create an NSX transport zone on any vCenter Server registered to VMware Cloud Director. See the *NSX Administration Guide*.

2 Create a `VMWNetworkPool` element that specifies the properties of the network pool.

You must use the NSX API to retrieve the required NSX transport zone information. See the request portion of Create a VXLAN-Backed Network Pool.

3 POST the `VMWNetworkPool` element you created in Step 2 to your cloud's `add` URL for `networkPools`.

See the request portion of Create a VXLAN-Backed Network Pool.

Example: Create a VXLAN-Backed Network Pool

This example creates a VXLAN-backed network pool. You must use the NSX API to retrieve the transport zone `MoRef` and `VimObjectType` that are required in the `TransportZoneRef`.

Request:

```
POST https://vcloud.example.com/api/admin/extension/networkPools
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vmext:VMWNetworkPool
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   name="Example VXLAN Pool"
   type="application/vnd.vmware.admin.networkPool xml">
   <vcloud:Description>Example VXLAN-backed network pool</vcloud:Description>
   <vmext:TransportZoneRef>
      <vmext:VimServerRef
         href="https://vcloud.example.com3/api/admin/extension/vimServer/9"
         type="application/vnd.vmware.admin.vmwvirtualcenter xml" />
      <vmext:MoRef>vdnscope-12</vmext:MoRef>
      <vmext:VimObjectType>VDN_SCOPE</vmext:VimObjectType>
   </vmext:TransportZoneRef>
</vmext:VMWNetworkPool>
```

The response includes a `Task` that tracks the creation of the network pool, and a set of `Link` elements that you can use to operate on or modify it.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<vmext:VMWNetworkPool
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   name="Example VXLAN Pool"
   type="application/vnd.vmware.admin.networkPool+xml"
   href="https://vcloud.example.com/api/admin/extension/networkPool/23" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.networkPool+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPool/23" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/networkPool/23" />
   <vcloud:Description>Example VXLAN-backed network pool</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         ...
         operation="Creating Network Pool 23"
         ...
      </vcloud:Task>
   </vcloud:Tasks>
   ...
</vmext:VMWNetworkPool>
```

Create a VLAN-Backed Network Pool

To create a VLAN-backed network pool, create a `VMWNetworkPool` element whose `type` attribute has the value `VlanPoolType`, and POST the element to your cloud's `add` link for `networkPools`.

A VLAN-backed network pool is backed by a range of VLAN IDs.

Prerequisites

- This operation is restricted to system administrators.
- Verify that you know your cloud's `add` URL for `networkPools`. See Create a Network Pool.
- Verify that at least one vCenter server attached to your cloud has network resources available. See Retrieve a List of Available Portgroups and Switches from a vCenter Server

Procedure

1 Choose a vCenter server to provide a switch for the network pool.

2 Create a `VMWNetworkPool` element that specifies the properties of the network pool.

See the request portion of Create a VLAN-Backed Network Pool.

3 POST the `VMWNetworkPool` element you created in Step 2 to your cloud's `add` URL for `networkPools`.

See the request portion of Create a VLAN-Backed Network Pool.

Example: Create a VLAN-Backed Network Pool

Use the query service to retrieve a list of DV Switch objects available on vCenter servers registered to this cloud.

```
https://vcloud.example.com/api/query?type=dvSwitch&format=records
```

The query response includes the values you'll need for the `VimServerRef` and `MoRef` elements. The `VimObjectType` for a DV Switch is always `DV_SWITCH`.

Request:

```
POST https://vcloud.example.com/api/admin/extension/networkPools
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="vmext:VlanPoolType"
   name="example-Vlan-pool">
   <vcloud:Description>Example VLAN-backed network pool</vcloud:Description>
   <vmext:VlanRange>
      <vmext:Start>1</vmext:Start>
      <vmext:End>4</vmext:End>
   </vmext:VlanRange>
   <vmext:VimSwitchRef>
      <vmext:VimServerRef
         href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
      <vmext:MoRef>dvs-33</vmext:MoRef>
      <vmext:VimObjectType>DV_SWITCH</vmext:VimObjectType>
   </vmext:VimSwitchRef>
</vmext:VMWNetworkPool>
```

The response includes a `Task` that tracks the creation of the network pool, and a set of `Link` elements that you can use to operate on or modify it.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xsi:type="vmext:VlanPoolType"
   name="example-Vlan-pool"
   id="urn:vcloud:networkpool:67"
   type="application/vnd.vmware.admin.networkPool+xml"
   href="https://vcloud.example.com/api/admin/extension/networkPool/67" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.networkPool+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPool/67" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/networkPool/67" />
   <vcloud:Description>Example VLAN-backed network pool</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         ...
         operation="Creating Network Pool 67"
         ...
      </vcloud:Task>
   </vcloud:Tasks>
   ...
</vmext:VMWNetworkPool>
```

Create a Portgroup-Backed Network Pool

To create a portgroup-backed network pool, you create a `VMWNetworkPool` element whose `type` attribute has the value `PortGroupPoolType`, and POST the element to your cloud's `add` link for `networkPools`.

Prerequisites

- This operation is restricted to system administrators.
- Verify that you know your cloud's `add` URL for `networkPools`. See Create a Network Pool.
- Verify that at least one vCenter server attached to your cloud has network resources available. See Retrieve a List of Available Portgroups and Switches from a vCenter Server

Procedure

1 Choose a vCenter server to provide a portgroup for the network pool.

2 Create a `VMWNetworkPool` element that specifies the properties of the network pool.

  See the request portion of Create a Portgroup-Backed Network Pool.

3 POST the `VMWNetworkPool` element you created in Step 2 to your cloud's `add` URL for `networkPools`. See Create a Network Pool.

  See the request portion of Create a Portgroup-Backed Network Pool.

Example: Create a Portgroup-Backed Network Pool

Request:

```
POST https://vcloud.example.com/api/admin/extension/networkPools
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="vmext:PortGroupPoolType"
   name="example-portgroup-pool"
   type="application/vnd.vmware.admin.networkPool+xml">
   <vcloud:Description>Example portgroup-backed network pool</vcloud:Description>
   <vmext:PortGroupRefs>
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>dvportgroup-32</vmext:MoRef>
         <vmext:VimObjectType>DV_PORTGROUP</vmext:VimObjectType>
      </vmext:VimObjectRef>
   </vmext:PortGroupRefs>
   <vmext:VimServer
      href="https://vcloud.example.com/api/admin/extension/vimServer/9"></vmext:VimServer>
</vmext:VMWNetworkPool>
```

The response includes a `Task` that tracks the creation of the network pool, and a set of `Link` elements that you can use to operate on or modify it.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.networkPool+xml
...
<vmext:VMWNetworkPool
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="vmext:PortGroupPoolType"
   name="example-portgroup-pool"
   id="urn:vcloud:networkpool:66"
   type="application/vnd.vmware.admin.networkPool+xml"
   href="https://vcloud.example.com/api/admin/extension/networkPool/66" ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwNetworkPoolReferences+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPoolReferences" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.networkPool+xml"
      href="https://vcloud.example.com/api/admin/extension/networkPool/66" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/networkPool/66" />
   <vcloud:Description>Example portgroup-backed network pool</vcloud:Description>
   <vcloud:Tasks>
      <vcloud:Task
         status="running"
         ...
         operation="Creating Network Pool 66"
         ...
      </vcloud:Task>
   </vcloud:Tasks>
   ...
</vmext:VMWNetworkPool>
```

Add a VDC to an Organization

A system administrator can allocate resources from a provider VDC to a VDC in an organization by POSTing a `CreateVdcParams` element to an organization's `add` URL for VDCs.

An organization virtual datacenter (organization VDC) is a deployment environment for virtual systems owned by the containing organization, and an allocation mechanism for resources such as networks, storage, CPU, and memory. In an organization VDC, computing resources are fully virtualized, and can be allocated based on demand, service level requirements, or a combination of the two.

Prerequisites

- This operation is restricted to system administrators.
- Retrieve the list of network pools. Several types of organization VDC networks require the VDC to include a network pool, which you can specify when you create or update the VDC. See Retrieve a List of External Networks and Network Pools for information about how to retrieve this list.
- If you want the new VDC to adopt specific resource pools, see Adopt Resource Pools With a VDC.

Procedure

1 Retrieve the XML representation of the organization to which you want to add the VDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26
```

2 Examine the response to locate the `Link` element that contains the URL for adding VDCs to the organization.

   This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.createVdcParams+xml`, as shown here:

```
<Link
   href="https://vcloud.example.com/api/admin/org/26/vdcsparams"
   rel="add"
   type="application/vnd.vmware.admin.createVdcParams+xml"/>
```

3 Choose a provider VDC to supply resources for the new organization VDC

   a Retrieve the XML representation of the `VCloud` object and examine the `ProviderVdcReferences` element it contains.

      The following request retrieves the representation of the `VCloud` object:

```
GET https://vcloud.example.com/api/admin
```

      The `VCloud` element contains a `ProviderVdcReferences` element. Each provider VDC in the system is represented in that element by a `ProviderVdcReference` element, as shown here:

```
<ProviderVdcReference
   type="application/vnd.vmware.admin.providervdc+xml"
   name="Main Provider"
   href="https://vcloud.example.com/api/admin/providervdc/2"/>
```

   b (Optional) List the organization VDCs that each `ProviderVdc` supports.

      The following request retrieves the list of organization VDCs that `.../providervdc/2` supports:

```
GET https://vcloud.example.com/api/admin/providervdc/2/vdcReferences
```

      Taking this optional step can help you allocate `ProviderVdc` resources equitably across the organization VDCs in a cloud.

4 Create a CreateVdcParams request body.

a Include an AllocationModel element that specifies how compute resources are allocated by this VDC.

Choose one of the following values for AllocationModel:

AllocationVApp

Pay as you go. Resources are committed to the organization VDC only when vApps are created in it. When you use this allocation model, any Limit values you specify for Memory and CPU are ignored when you create a vApp and returned as 0 when you retrieve a vApp. Resources available to this kind of organization VDC can grow or shrink as needed when its provider VDC has multiple resource pools.

AllocationPool

Only a percentage of the resources you allocate are committed to the organization VDC

ReservationPool

All the resources you allocate are committed as a pool to the organization VDC. vApps in VDCs that support this allocation model can specify values for resources and limitations.

Flex

You can control the resource consumption at both the VDC and the individual virtual machine levels. The flex allocation model supports the capabilities of organization VDC compute policies. Flex allocation model supports all allocation configurations that are available in the other allocation models. When you use this allocation model, you include the IsElasticVDC and IncludeMemoryOverhead elements. If you want the VDC to be elastic, you must set the IsElasticVDC value to true. To include memory overhead into the tenant quota, you must set the IncludeMemoryOverhead value to true.

Note If you choose AllocationPool or ReservationPool, you can also include an OverCommitAllowed element in the CreateVdcParams request. Setting its value to false prevents creation of the VDC if the ComputeCapacity you specified is greater than what the backing Provider VDC can supply.

b Include at least one VdcStorageProfile element that specifies a ProviderVdcStorageProfile defined in the Provider VDC you chose in Step 3.

c Include a NetworkPoolReference element.

The VDC must include a network pool if you want to create routed or isolated networks in it.

d Include a ProviderVdcReference element that contains a reference to the Provider VDC you chose in Step 3.

See the request portion of Create an Organization VDC with Pay As You Go Reservation Allocation Model.

5 POST the `CreateVdcParams` request body to the organization's `add` link for `vdcs`.

   See the request portion of Create an Organization VDC with Pay As You Go Reservation Allocation Model.

Results

The server creates the new VDC in the specified organization and returns an `AdminVdc` element that includes a set of `Link` elements that you can use to access, remove, or modify the new VDC. Users can reference this VDC using the URL specified in the `href` attribute in the `Link` where `rel="alternate"`. See the response portion of Create an Organization VDC with Pay As You Go Reservation Allocation Model.

If the target organization already contains the maximum number of VDCs allowed by the system administrator, the request fails. A system administrator can change the VDC quota for an organization by updating the value of `VdcQuota` in the organization's `GeneralOrgSettings`.

Example: Create an Organization VDC with Pay As You Go Reservation Allocation Model

This example adds an `AllocationvApp` VDC to the organization created in Create an Organization. The new organization VDC is provisioned from the Provider VDC created in Create a Provider VDC, and includes a storage profile named `Silver`, which is backed by one of the storage profiles available in the Provider VDC. It also includes a network pool, so that it is capable of supporting routed and isolated organization VDC networks. See Retrieve a List of External Networks and Network Pools for information on how to find a `NetworkPoolReference` to use.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/vdcsparams
Content-Type: application/vnd.vmware.admin.createVdcParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CreateVdcParams
   name="org26vdc1"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Example VDC</Description>
   <AllocationModel>AllocationVApp</AllocationModel>
   <ComputeCapacity>
      <Cpu>
         <Units>MHz</Units>
         <Allocated>2048</Allocated>
         <Limit>2048</Limit>
      </Cpu>
      <Memory>
         <Units>MB</Units>
         <Allocated>2048</Allocated>
         <Limit>2048</Limit>
      </Memory>
   </ComputeCapacity>
   <NicQuota>0</NicQuota>
   <NetworkQuota>100</NetworkQuota>
   <VdcStorageProfile>
      <Enabled>true</Enabled>
      <Units>MB</Units>
      <Limit>20480</Limit>
      <Default>true</Default>
      <ProviderVdcStorageProfile
         href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101" />
   </VdcStorageProfile>
   <ResourceGuaranteedMemory>1</ResourceGuaranteedMemory>
   <ResourceGuaranteedCpu>1</ResourceGuaranteedCpu>
   <VCpuInMhz>2048</VCpuInMhz>
   <IsThinProvision>false</IsThinProvision>
   <NetworkPoolReference
      href="https://vcloud.example.com/api/admin/extension/networkPool/313"/>
   <ProviderVdcReference
      name="Main Provider"
      href="https://vcloud.example.com/api/admin/providervdc/35" />
   <UsesFastProvisioning>true</UsesFastProvisioning>
</CreateVdcParams>
```

The response, a subset of which appears here, contains information extracted from the request, and includes a `Task` element that tracks creation of the VDC. The response also includes `Link` elements that enable administrative operations on the VDC, and a `Capabilities` element that lists the VMware virtual hardware architectures that the VDC supports. These elements are retrieved from the Provider VDC that you specified when you created the `CreateVdcParams`. While the VDC is under construction, its `status` remains `0`.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.vdc+xml
...
<AdminVdc
   xmlns="http://www.vmware.com/vcloud/v1.5"
   status="0"
   name="org26vdc1"
   id="urn:vcloud:vdc:44"
   type="application/vnd.vmware.admin.vdc+xml"
   href="https://vcloud.example.com/api/admin/vdc/44" ... >
   <Link
      rel="up"
      type="application/vnd.vmware.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/26" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/vdc/44/metadata" />
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.vdc+xml"
      href="https://vcloud.example.com/api/vdc/44" />
   <Link
      rel="down"
      type="application/vnd.vmware.admin.OrganizationVdcResourcePoolSet+xml"
      href="https://vcloud.example.com/api/admin/extension/vdc/44/resourcePools" />
   <Description>Example VDC</Description>
   ...
   <Tasks>
      <Task
         name="task"
         status="running"
         operation="Creating Virtual Datacenter org26vdc1(44)"
         ...
      </Task>
   </Tasks>
   <AllocationModel>AllocationVApp</AllocationModel>
   ...
   <Capabilities>
      <SupportedHardwareVersions>
         <SupportedHardwareVersion>vmx-04</SupportedHardwareVersion>
         <SupportedHardwareVersion>vmx-08</SupportedHardwareVersion>
      </SupportedHardwareVersions>
   </Capabilities>
   ...
</AdminVdc>
```

When construction is complete, the `status` changes to `1` and the `Task` is no longer included in representation. The following changes in the `AdminVdc` are also evident:

- A reference to the vSphere resource pool that supports the VDC appears in a `ResourcePoolRefs` element and, for compatibility, in a `VCloudExtension` element.
- There is an empty `ResourceEntities` element, because the VDC contains no `Media`, `VAppTemplate`, or `Disk` entities. For information about adding them, see Chapter 4 Provisioning an Organization.
- There is an empty `AvailableNetworks` element. To add networks to this organization VDC, see Create an Organization VDC Network.
- Additional `Link` elements are included for operations that are now valid, but that were not valid while the VDC was under construction.

```
<AdminVdc
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   status="1"
   name="org26vdc1"
   id="urn:vcloud:vdc:44"
   type="application/vnd.vmware.admin.vdc+xml"
   href="https://vcloud.example.com/api/admin/vdc/44" ... >
   <VCloudExtension
      required="false">
      <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="vc1"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-949</vmext:MoRef>
```

```
          <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
       </vmext:VimObjectRef>
    </VCloudExtension>
      <Link
       rel="up"
       type="application/vnd.vmware.admin.organization+xml"
       href="https://vcloud.example.com/api/admin/org/e0b93bca-5dc2-453c-b3dc-bba8067d32b6" />
    <Link
       rel="edit"
       type="application/vnd.vmware.admin.vdc+xml"
       href="https://vcloud.example.com/api/admin/vdc/44" />
    <Link
       rel="disable"
       href="https://vcloud.example.com/api/admin/vdc/44/action/disable" />
    <Link
       rel="down"
       type="application/vnd.vmware.vcloud.metadata+xml"
       href="https://vcloud.example.com/api/admin/vdc/44/metadata" />
    <Link
       rel="alternate"
       type="application/vnd.vmware.vcloud.vdc+xml"
       href="https://vcloud.example.com/api/vdc/44" />
    <Link
       rel="add"
       type="application/vnd.vmware.admin.edgeGateway+xml"
       href="https://vcloud.example.com/api/admin/vdc/44/edgeGateways" />
    <Link
       rel="edgeGateways"
       type="application/vnd.vmware.vcloud.query.records+xml"
       href="https://vcloud.example.com/api/admin/vdc/44/edgeGateways" />
    <Link
       rel="add"
       type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
       href="https://vcloud.example.com/api/admin/vdc/44/networks" />
    <Link
       rel="orgVdcNetworks"
       type="application/vnd.vmware.vcloud.query.records+xml"
       href="https://vcloud.example.com/api/admin/vdc/44/networks" />
    <Link
       rel="down"
       type="application/vnd.vmware.admin.OrganizationVdcResourcePoolSet+xml"
       href="https://vcloud.example.com/api/admin/extension/vdc/44/resourcePools" />
    <Link
       rel="edit"
       type="application/vnd.vmware.admin.updateVdcStorageProfiles+xml"
       href="https://vcloud.example.com/api/admin/vdc/44/vdcStorageProfiles" />
    ...
    <ResourceEntities />
    <AvailableNetworks />
    ...
    <VdcStorageProfiles>
       <VdcStorageProfile
          type="application/vnd.vmware.admin.vdcStorageProfile+xml"
          name="Silver"
          href="https://vcloud.example.com/api/admin/vdcStorageProfile/158" />
```

```
   </VdcStorageProfiles>
   ...
   <ResourcePoolRefs>
       <vmext:VimObjectRef>
         <vmext:VimServerRef
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml"
            name="vc1"
            href="https://vcloud.example.com/api/admin/extension/vimServer/9" />
         <vmext:MoRef>resgroup-949</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:VimObjectRef>
    </ResourcePoolRefs>
</AdminVdc>
```

Adopt Resource Pools With a VDC

When adding a VDC to an organization, a system administrator can specify one or more resource pools that are available in the supporting Provider VDC. When the VDC is created, the specified resource pools are said to have been adopted by the VDC. Any vCenter VMs that exist in an adopted resource pool become available as discovered vApps in the new VDC.

When you create an organization VDC without specifying a resource pool, the system creates the VDC using the default resource pool of the specified Provider VDC. Before creating an organization VDC, you can query a Provider VDC to get a list of all resource pools that are candidates for adoption by the new VDC. If any resource pools contain VMs created in vCenter that your organization would like to access in a VDC, specify those pools when you create the VDC. vCenter VMs in the specified pools appear in the new VDC as discovered vApps and are candidates for adoption.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Choose a provider VDC to supply resources for the new organization VDC

   a Retrieve the XML representation of the `VCloud` object and examine the `ProviderVdcReferences` element it contains.

   The following request retrieves the representation of the `VCloud` object:

   ```
   GET https://vcloud.example.com/api/admin
   ```

   The `VCloud` element contains a `ProviderVdcReferences` element. Each provider VDC in the system is represented in that element by a `ProviderVdcReference` element, as shown here:

   ```
   <ProviderVdcReference
      type="application/vnd.vmware.admin.providervdc+xml"
      name="Main Provider"
      href="https://vcloud.example.com/api/admin/providervdc/2"/>
   ```

   b (Optional) List the organization VDCs that each `ProviderVdc` supports.

   The following request retrieves the list of organization VDCs that `.../providervdc/2` supports:

   ```
   GET https://vcloud.example.com/api/admin/providervdc/2/vdcReferences
   ```

   Taking this optional step can help you allocate `ProviderVdc` resources equitably across the organization VDCs in a cloud.

2 Retrieve the XML representation of the provider VDC you have chosen to back the new organization VDC.

The `VMWProvderVDC` response body includes a `Link` of the following form:

```
<Link
   rel="down" href="https://vcloud.example.com/api/admin/extension/providervdc/2/
discoverResourcePools"
   type="application/vnd.vmware.admin.vmwDiscoveredResourcePools+xml"/>"
```

3 Retrieve the list of resource pools that are candidates for adoption by a VDC backed by the selected Provider VDC.

Use the `Link` shown in Step 3 to make a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/2/discoverResourcePools
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWDiscoveredResourcePools
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:VMWDiscoveredResourcePool
      name="pvdc-1 (mapped as pVDC)"
```

```
      valideCandidate="false">
      <vcloud:Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/providervdc/35/
discoverResourcePools/resgroup-80"
         type="application/vnd.vmware.admin.vmwDiscoveredResourcePools+xml" />
      <vmext:ResourcePoolVimObjectRef>
         <vmext:VimServerRef
            href="https://vcloud.example.com/api/admin/extension/vimServer/3"
            name="vCenter System 1"
            type="application/vnd.vmware.admin.vmwvirtualcenter+xml" />
         <vmext:MoRef>resgroup-80</vmext:MoRef>
         <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
      </vmext:ResourcePoolVimObjectRef>
   </vmext:VMWDiscoveredResourcePool>
</vmext:VMWDiscoveredResourcePools>
```

4 Create a `CreateVdcParams` request body.

   See the request portion of Add a VDC With an Adopted Resource Pool to an Organization.

5 POST the `CreateVdcParams` request body to the organization's `add` link for `vdcs`.

   See the request portion of Add a VDC With an Adopted Resource Pool to an Organization.

Example: Add a VDC With an Adopted Resource Pool to an Organization

This example modifies the request shown in Create an Organization VDC with Pay As You Go Reservation Allocation Model to add a `ResourcePoolRefs` element that specifies the resource pool identified in Step 3.

Request:

```
POST https://vcloud.example.com/api/admin/org/26/vdcsparams
Content-Type: application/vnd.vmware.admin.createVdcParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<CreateVdcParams
   name="org26vdc1"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Example VDC</Description>
   <AllocationModel>AllocationVApp</AllocationModel>
   <ComputeCapacity>
      <Cpu>
         <Units>MHz</Units>
         <Allocated>2048</Allocated>
         <Limit>2048</Limit>
      </Cpu>
      <Memory>
         <Units>MB</Units>
         <Allocated>2048</Allocated>
         <Limit>2048</Limit>
      </Memory>
   </ComputeCapacity>
   <NicQuota>0</NicQuota>
   <NetworkQuota>100</NetworkQuota>
   <VdcStorageProfile>
      <Enabled>true</Enabled>
      <Units>MB</Units>
      <Limit>20480</Limit>
      <Default>true</Default>
      <ProviderVdcStorageProfile
         href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101" />
   </VdcStorageProfile>
   <ResourceGuaranteedMemory>1</ResourceGuaranteedMemory>
   <ResourceGuaranteedCpu>1</ResourceGuaranteedCpu>
   <VCpuInMhz>2048</VCpuInMhz>
   <IsThinProvision>false</IsThinProvision>
   <NetworkPoolReference
      href="https://vcloud.example.com/api/admin/extension/networkPool/313"/>
   <ProviderVdcReference
      name="Main Provider"
      href="https://vcloud.example.com/api/admin/providervdc/35" />
   <ResourcePoolRefs>
        <vmext:VimObjectRef >
            <vmext:VimServerRef href="https://vcloud.example.com/api/admin/extension/
vimServer/36"
             name="VC"
             type="application/vnd.vmware.admin.vmwvirtualcenter+xml"/>
            <vmext:MoRef>resgroup-70</vmext:MoRef>
            <vmext:VimObjectType>RESOURCE_POOL</vmext:VimObjectType>
        </vmext:VimObjectRef>
    </ResourcePoolRefs>
   <UsesFastProvisioning>true</UsesFastProvisioning>
</CreateVdcParams>
```

Configure Storage I/O Control Support in an Organization VDC

You can modify a VDC storage profile to enable support for vCenter Storage I/O Control if that feature is configured in the underlying Provider VDC.

Managed read/write performance in physical storage devices and virtual disks is defined in units called IOPS, which measure read/write operations per second. When an organization VDC storage profile is backed by a Provider VDC storage profile that includes storage devices that are capable of IOPS allocation, you can configure disks that use it to request a specified level of I/O performance. A storage profile configured with IOPS support delivers its default IOPS value to all disks that use it, even disks that are not configured to request a specific IOPS value. A hard disk configured to request a specific IOPS value cannot use a storage profile whose maximum IOPS value is lower than the requested value, or a storage profile that is not configured with IOPS support.

VMware Cloud Director sets an IOPS limit and reservation for every disk that uses an IOPS-enabled storage profile. vSphere is responsible for allocating the IOPS capacity of the underlying datastore across all virtual disks that use the storage profile. IOPS management is primarily intended to ensure that no disk can consume more than its fair share of IOPS. Realized IOPS for a given disk are limited by what the backing LUN can provide, and can be influenced by factors such as read/write block size. While a given storage profile can include a mix of datastores that IOPS-enabled and those that are not, such configurations can interfere with the system's ability to allocate IOPS fairly across all disks that use the storage profile.

The ability of a Provider VDC storage profile to provide IOPS support depends on the configuration of the vSphere datastores that support it. See Configure Storage I/O Control Support in a Provider VDC.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the VDC in the admin view, and examine the `AdminVdc` element to find the `VdcStorageProfiles` element it contains.

   Request:

   ```
   GET https://vcloud.example.com/api/admin/vdc/44
   ```

   Response:

   ```
   <AdminVdc>
      ...
      <VdcStorageProfiles>
         <VdcStorageProfile
            href="https://vcloud.example.com/api/admin/vdcStorageProfile/99"
            name="*"
            type="application/vnd.vmware.admin.vdcStorageProfile+xml"/>
         <VdcStorageProfile
            href="https://vcloud.example.com/api/admin/vdcStorageProfile/128"
            name="bronze"
            type="application/vnd.vmware.admin.vdcStorageProfile+xml"/>
      </VdcStorageProfiles>
      ...
   </AdminVdc>
   ```

2 Retrieve an organization VDC storage profile to discover the provider VDC storage profile that backs it.

   Request:

   ```
   GET https://vcloud.example.com/api/admin/vdcStorageProfile/128
   ```

   Response:

   ```
   <AdminVdcStorageProfile
      xmlns="http://www.vmware.com/vcloud/v1.5"
      name="bronze"
      type="application/vnd.vmware.admin.vdcStorageProfile+xml" ... >
      ...
   ```

```
   <ProviderVdcStorageProfile
      href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101"
      name="Bronze"
      type="application/vnd.vmware.admin.pvdcStorageProfile+xml"/>
</AdminVdcStorageProfile>
```

3 Verify that the provider VDC storage profile can provide IOPS support.

   You can use `providerVdcStorageProfile` query as shown here, and include a filter expression that specifies the `href` value of the `ProviderVdcStorageProfile`. Examine the `iopsAllocated` and `iopsCapacity` attributes in the response. Values greater than `0` indicate that the provider VDC storage profile can provide IOPS support:

```
GET https://vcloud.example.com/api/query?type=providerVdcStorageProfile&format=records\
&filter=href==https://vcloud.example.com/api/admin/pvdcStorageProfile/101
...

<QueryResultRecords
...
   <ProviderVdcStorageProfileRecord
      iopsAllocated="256000"
      iopsCapacity="400000"
   ... />
</QueryResultRecords>
```

4 Make a PUT request to the `rel="edit"` link from the `AdminVdcStorageProfile` you retrieved in Step 2 and supply the modified `AdminVdcStorageProfile` as the request body.

   Supply the `Vdc` element as the request body.

Example: Add IOPS Support to an Organization VDC Storage Profile

This request adds IOPS support to a storage profile.

Request:

```
PUT https://vcloud.example.com/api/admin/vdcStorageProfile/128
Content-Type: application/vnd.vmware.admin.vdcStorageProfile+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<AdminVdcStorageProfile
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="bronze"
   type="application/vnd.vmware.admin.vdcStorageProfile+xml">
   <Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/vdcStorageProfile/128"
      type="application/vnd.vmware.vcloud.vdcStorageProfile+xml" />
   <Enabled>true</Enabled>
   <Units>MB</Units>
   <Limit>1024</Limit>
   <Default>true</Default>
   <IopsSettings>
      <Enabled>true</Enabled>
      <DiskIopsMax>4000</DiskIopsMax>
      <DiskIopsDefault>1000</DiskIopsDefault>
      <StorageProfileIopsLimit>200000</StorageProfileIopsLimit>
      <DiskIopsPerGbMax>100</DiskIopsPerGbMax>
   </IopsSettings>

   <ProviderVdcStorageProfile
      href="https://vcloud.example.com/api/admin/pvdcStorageProfile/101"
      name="Bronze"
      type="application/vnd.vmware.admin.pvdcStorageProfile+xml" />
</AdminVdcStorageProfile>
```

If the storage profile contains storage pods, this request fails with a message like this one:

```
BadRequestException: Storage profile n contains storage pod p. Enabing IOPS guarantee is not
supported on a storage profile containing storage pods.
```

This request fails with a `BadRequestException` if any of the following are true:

- The storage profile includes one or more storage pods.
- The storage profile includes one or more VMware Virtual SAN datastores.
- The value you specified for `DiskIopsMax`, `DiskIopsPerGbMax`, or `DiskIopsDefault` is not in the range allowed by the storage profile.

The response is a `Task`.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
   ...
</Task>
```

Working with NSX Data Center for vSphere Edge Clusters

To isolate the compute workloads from the networking workloads, VMware Cloud Director supports the edge cluster object. An edge cluster consists of a vSphere resource pool and a storage policy that are used only for organization VDC edge gateways. Provider virtual data centers cannot use resources dedicated to edge clusters, and edge clusters cannot use resources dedicated to provider virtual data centers.

Edge clusters provide a dedicated L2 broadcast domain, which reduces the VLAN sprawls and ensures the network security and isolation. For example, the edge cluster can contain additional VLANs for peering with physical routers.

You can create any number of edge clusters. You can assign an edge cluster to an organization VDC as a primary or secondary edge cluster.

- The primary edge cluster for an organization VDC is used for the main edge appliance of an organization VDC edge gateway.
- The secondary edge cluster for an organization VDC is used for the standby edge appliance when an edge gateway is in HA mode.

Different organization VDCs can share edge clusters or can have their own dedicated edge clusters.

Starting with vCloud Director 9.7, the old process for using metadata to control the edge gateway placement is deprecated. See https://kb.vmware.com/kb/2151398.

You can migrate legacy edge gateways to newly created edge clusters by redeploying these edge gateways.

Preparing Your Environment for an Edge Cluster

1 In vSphere, create the resource pool for the target edge cluster.

  If an organization virtual data center is using a VLAN network pool, the VLAN network pool and the edge cluster for this organization virtual data center must reside on the same vSphere distributed switch.

2 If an organization virtual data center is using a VXLAN network pool, in NSX, add the edge cluster to the VXLAN transport zone, after which synchronize the VXLAN network pool in VMware Cloud Director.

3 In vSphere, create the edge cluster storage profile.

Creating and Managing Edge Clusters

After you prepare your environment, to create and manage edge clusters, you must use the VMware Cloud Director OpenAPI `EdgeClusters` methods. See *Getting Started with VMware Cloud Director OpenAPI* at https://developer.vmware.com/.

Viewing edge clusters requires the **Edge Cluster View** right. Creating, updating, and deleting edge clusters require the **Edge Cluster Manage** right.

When you create an edge cluster, you specify the name, the vSphere resource pool, and the storage profile name.

After you create an edge cluster, you can modify its name and description. After you delete or move its containing edge gateways, you can delete an edge cluster.

Assigning an Edge Cluster to an Organization VDC

After you create an edge cluster, you can assign this edge cluster to an organization VDC by updating the organization VDC network profile. You can assign an edge cluster to an organization VDC as a primary or secondary edge cluster.

If you do not assign a secondary edge cluster, the standby edge appliance of an edge gateway in HA mode is deployed on the primary edge cluster but on a host different from the host running the primary edge appliance.

To update, view, and delete organization VDC network profiles, you must use the VMware Cloud Director OpenAPI `VdcNetworkProfile` methods. See *Getting Started with VMware Cloud Director OpenAPI* at https://developer.vmware.com/.

Considerations:

- The primary and secondary edge clusters must reside on the same vSphere distributed switch.
- If the organization VDC uses a VXLAN network pool, the NSX Transport Zone must span the compute and the edge clusters.
- If the organization VDC uses a VLAN network pool, the edge clusters and the compute clusters must be on the same vSphere distributed switch.

If you update again the primary or secondary edge cluster of an organization VDC, to move an existing edge gateway to the new cluster, you must redeploy this edge gateway.

Create an Edge Gateway

An Edge Gateway is a virtual router for organization VDC networks. You can configure it to provide network services such as DHCP, firewall, NAT, static routing, VPN, and load balancing.

Starting with vCloud Director 9.7, the compute workload and the networking workload are isolated by using different vSphere resource pools and storage policies. Edge Gateways reside on edge clusters that you must previously create. See Working with NSX Data Center for vSphere Edge Clusters.

You can migrate legacy edge gateways to the corresponding edge clusters by redeploying these edge gateways. See .

Important Starting with version 9.7, VMware Cloud Director supports only advanced edge gateways. You must convert any legacy non-advanced edge gateway to an advanced gateway. See https://kb.vmware.com/kb/66767.

An Edge Gateway can support up to ten interfaces. These interfaces are categorized as uplinks when they connect to an external network, and internal interfaces when they connect to an organization VDC network. You must specify at least one uplink interface when you create an Edge Gateway. All uplink interfaces on an Edge Gateway must connect to an external network available in the Provider VDC that backs the organization VDC in which you are creating the Edge Gateway. Internal interfaces are created automatically when you create a routed organization VDC network that connects to an Edge Gateway.

Prerequisites

- This operation is restricted to system administrators.

- An Edge Gateway requires an organization VDC backed by a Provider VDC that contains at least one external network.
- If you want to deploy the Edge Gateway on a dedicated edge cluster, create and assign an edge cluster to the organization virtual data center. See Working with NSX Data Center for vSphere Edge Clusters.

Procedure

1 Choose an organization VDC to contain the Edge Gateway.

2 Choose an external network to use for the Edge Gateway's initial uplink interface.

   This external network must be one of the networks listed in the `AvailableNetworks` element of the Provider VDC that backs the organization VDC in which you are creating the Edge Gateway. Follow these steps to find a suitable external network.

   a Retrieve the XML representation of the organization VDC in which you are creating the Edge Gateway.

      Use a request like this one:

```
GET https://vcloud.example.com/api/admin/vdc/44
```

      The `ProviderVdcReference` element in the response contains a reference to the Provider VDC that backs this organization VDC.

```
<AdminVdc ...>
   ...
   <ProviderVdcReference
      type="application/vnd.vmware.admin.providervdc+xml"
      name="PVDC-Example"
      href="https://vcloud.example.com/api/admin/extension/providervdc/35"
   ...
</AdminVdc>
```

   b Retrieve the the XML representation of the Provider VDC.

      Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

      The `AvailableNetworks` element in the response lists the external networks that are available to that Provider VDC, and to all the organization VDCs that it supports.

```
<vmext:VMWProviderVdc ... >
   ...
   <vcloud:AvailableNetworks>
      <vcloud:Network
         type="application/vnd.vmware.admin.network+xml"
         name="VC1-VLAN48"
         href="https://vcloud.example.com/api/admin/network/297" />
      <vcloud:Network ... />
      <vcloud:Network ... />
   </vcloud:AvailableNetworks>
   ...
</vmext:VMWProviderVdc>
```

      Choose an available external network to provide the initial interface for the new Edge Gateway. See Create an Edge Gateway for more information about criteria for choosing an external network.

3. Create an `EdgeGateway` element.

   In the `GatewayInterfaces` element, create a `GatewayInterface` element that defines an uplink interface.

   - Specify `uplink` for the `InterfaceType`.
   - Include the external network reference you retrieved in Step 2 in the `Network` element.

     If you assigned an edge cluster to the organization VDC, you must select external networks that are accessible to this cluster.
   - If you plan to create a NAT service or load balancer service in the Edge Gateway, you must create an IP sub-allocation for the uplink by including a `SubnetParticipation` element in the `GatewayInterface` element. IP addresses in the range you specify in this element are reserved for use by this Edge Gateway.

   For information about additional elements that an `EdgeGateway` can contain, see Create an Edge Gateway and the schema reference.

4. POST the `EdgeGateway` element to the URL for adding Edge Gateways to the organization VDC.

Results

The server takes the requested action and returns an XML representation of the partially-created object. This representation includes an `href` attribute, properties specified in the creation request, and an embedded `Task` element that tracks the creation of the object. When the task completes, the object has been created, and you can use the value of the `href` attribute with a GET request to retrieve the XML representation of the object.

See the response portion of Create an Edge Gateway.

Example: Create an Edge Gateway

This example adds an Edge Gateway to the organization VDC created in Add a VDC to an Organization . The uplink interface specifies one of the networks shown in 2.b.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/edgeGateways
Content-Type: application/vnd.vmware.admin.edgeGateway+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<EdgeGateway
   name="theEdge"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Example Edge Gateway</Description>
   <Configuration>
      <GatewayBackingConfig>compact</GatewayBackingConfig>
      <GatewayInterfaces>
         <GatewayInterface>
            <Name>uplink1</Name>
            <DisplayName>uplink1</DisplayName>
            <Network href="https://vcloud.example.com/api/admin/network/297" />
            <InterfaceType>uplink</InterfaceType>
            <SubnetParticipation>
               <Gateway>10.147.115.190</Gateway>
               <Netmask>255.255.255.0</Netmask>
            </SubnetParticipation>
         </GatewayInterface>
      </GatewayInterfaces>
      <HaEnabled>false</HaEnabled>
      <UseDefaultRouteForDnsRelay>false</UseDefaultRouteForDnsRelay>
   </Configuration>
</EdgeGateway>
```

The response is an `EdgeGateway` element with an embedded `Task` element that tracks the creation of the Edge Gateway object.

The response includes a number of `Link` elements that you can use to manage the new Edge Gateway. It also includes an `EdgeGatewayServiceConfiguration` element that contains a simple `FirewallService`, which drops all incoming and outgoing packets, effectively blocking all traffic through the Edge Gateway. This service is created by default if you do not specify an `EdgeGatewayServiceConfiguration` when you create the `EdgeGateway`. To remove or modify it, see Configure Edge Gateway Services.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<EdgeGateway
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="theEdge"
   id="urn:vcloud:gateway:2000"
   href="https://vcloud.example.com/api/admin/edgeGateway/2000" ... >
   <Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000"
      type="application/vnd.vmware.admin.edgeGateway+xml" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000" />
   <Link
      rel="up"
      href="https://vcloud.example.com/api/admin/vdc/44"
      type="application/vnd.vmware.admin.vdc+xml" />
   <Link
      rel="edgeGateway:redeploy"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/redeploy" />
   <Link
      rel="edgeGateway:configureServices"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/configureServices"
      type="application/vnd.vmware.admin.edgeGatewayServiceConfiguration+xml" />
   <Link
      rel="edgeGateway:configureSyslogServerSettings"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/
configureSyslogServerSettings"
      type="application/vnd.vmware.vcloud.SyslogSettings+xml" />
   <Link
      rel="edgeGateway:reapplyServices"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/reapplyServices" />
   <Link
      rel="edgeGateway:syncSyslogSettings"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/
syncSyslogServerSettings" />
   <Link
      rel="edgeGateway:upgrade"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/upgradeConfig" />
   <Link
      rel="edgeGateway:modifyFormFactor"
      href="https://vcloud.example.com/api/admin/edgeGateway/2000/action/modifyFormFactor"
      type="application/vnd.vmware.vcloud.edgeGatewayFormFactor+xml" />
   <Description>Example Edge Gateway</Description>
   <Tasks>
      <Task
         ...
         operation="Creating EdgeGateway theEdge(2000)"
         operationName="networkEdgeGatewayCreate"
         serviceNamespace="com.vmware.vcloud"
         ... >
         .
         .
         .
       </Task>
   </Tasks>
   <Configuration>
      <GatewayBackingConfig>compact</GatewayBackingConfig>
      <GatewayInterfaces>
         <GatewayInterface>
            <Network
               href="https://vcloud.example.com/api/admin/network/297"
               name=""
               type="application/vnd.vmware.admin.network+xml" />
            <InterfaceType>uplink</InterfaceType>
            <SubnetParticipation>
               <Gateway>10.147.115.190</Gateway>
               <Netmask>255.255.255.0</Netmask>
               <UseForDefaultRoute>false</UseForDefaultRoute>
            </SubnetParticipation>
            <ApplyRateLimit>false</ApplyRateLimit>
            <UseForDefaultRoute>false</UseForDefaultRoute>
         </GatewayInterface>
      </GatewayInterfaces>
      <EdgeGatewayServiceConfiguration>
         <FirewallService>
            <IsEnabled>true</IsEnabled>
            <DefaultAction>drop</DefaultAction>
            <LogDefaultAction>false</LogDefaultAction>
         </FirewallService>
      </EdgeGatewayServiceConfiguration>
      <HaEnabled>false</HaEnabled>
      <UseDefaultRouteForDnsRelay>false</UseDefaultRouteForDnsRelay>
      <AdvancedNetworkingEnabled>false</AdvancedNetworkingEnabled>
   </Configuration>
</EdgeGateway>
```

Create an Organization VDC Network With a Direct Connection

An organization VDC network with a direct connection provides direct layer 2 connectivity to machines and networks outside of the organization VDC. Machines outside of this organization VDC can connect to machines within the organization VDC directly.

An organization VDC network with a direct connection is configured as a child network of one of the external networks provisioned to the cloud by the system administrator.

Prerequisites

- This operation is restricted to system administrators.
- Retrieve the list of external networks. For information about how to retrieve this list, see External Networks and Network Pools.

Procedure

1 Choose the external network to which this organization VDC network will connect.

This external network must be one of the ones listed in the `AvailableNetworks` element of the Provider VDC that backs the organization VDC in which you are creating the new network. Follow these steps to find a suitable external network.

a Retrieve the XML representation of the organization VDC in which you are creating the new network.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/vdc/44
```

The `ProviderVdcReference` element in the response contains a reference to the Provider VDC that backs this organization VDC.

```
<AdminVdc ...>
   ...
   <ProviderVdcReference
         type="application/vnd.vmware.admin.providervdc+xml"
         name="PVDC-Example"
         href="https://vcloud.example.com/api/admin/extension/providervdc/35"
...
</AdminVdc>
```

b Retrieve the the XML representation of the Provider VDC.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

The `AvailableNetworks` element in the response lists the external networks that are available to that Provider VDC, and to all the organization VDCs that it supports.

```
<vcloud:AvailableNetworks>
   <vcloud:Network
      type="application/vnd.vmware.admin.network+xml"
      name="VC1-VLAN48"
      href="https://vcloud.example.com/api/admin/network/297" />
   <vcloud:Network ... />
   <vcloud:Network ... />
</vcloud:AvailableNetworks>
```

2 Create an `OrgVdcNetwork` element.

   a Specify the `href` of the external network you chose in Step 1 in the `ParentNetwork` element.

      The `type` and `name` attributes are optional here.

   b Set the `FenceMode` to `bridged`.

      This creates a direct connection to the parent network.

   See the request portion of Create an Organization VDC Network With a Direct Connection.

3 POST the `OrgVdcNetwork` element you created in Step 2 to the URL for adding networks to the organization VDC.

   See the request portion of Create an Organization VDC Network With a Direct Connection.

Results

The server takes the requested action and returns an XML representation of the partially-created object. This representation includes an `href` attribute, properties specified in the creation request, and an embedded `Task` element that tracks the creation of the object. When the task completes, the object has been created, and you can use the value of the `href` attribute with a GET request to retrieve the XML representation of the object.

See the response portion of Create an Organization VDC Network With a Direct Connection.

Example: Create an Organization VDC Network With a Direct Connection

This example adds a directly-connected network to the organization VDC created in Add a VDC to an Organization . Because the network has a `Configuration` whose `ParentNetwork` element specifies an external network to connect to and sets the `FenceMode` to `bridged`, it provides a direct connection to the parent network.

Request:

```
POST https://vcloud.example.com/api/admin/vdc/44/networks
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<OrgVdcNetwork
   name="Internet"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <Description>Bridged to the public Internet</Description>
   <Configuration>
      <ParentNetwork
         href="https://vcloud.example.com/api/admin/network/297" />
      <FenceMode>bridged</FenceMode>
   </Configuration>
</OrgVdcNetwork>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.orgVdcNetwork+xml
...
<OrgVdcNetwork
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="Internet"
   type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
   href="https://vcloud.example.com/api/admin/network/54" ...>
   <Link
      rel="edit"
      type="application/vnd.vmware.vcloud.orgVdcNetwork+xml"
      href="https://vcloud.example.com/api/admin/network/54" />
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/network/54" />
   <Link
      rel="up"
      type="application/vnd.vmware.admin.vdc+xml"
      href="https://vcloud.example.com/api/admin/vdc/44" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/admin/network/54/metadata" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.allocatedNetworkAddress+xml"
      href="https://vcloud.example.com/api/admin/network/54/allocatedAddresses/" />
   <Description>Bridged to the public Internet</Description>
   <Tasks>
      <Task
         name="task"
         status="running"
         operation="Creating Network Internet(54)"
         ...
      </Task>
   </Tasks>
   <Configuration>
      ...
   </Configuration>
   <IsShares>false</IsShared>
</OrgVdcNetwork>
```

Managing VM-Host Affinity Rules

A VMware Cloud Director system administrator can create groups of VMs in a resource pool, then use VM-Host affinity rules to specify whether members of a VM group should be deployed on members of a vSphere host DRS Group.

VMware Cloud Director VM-Host affinity rules provide VMware Cloud Director system administrators with a way to specify how vSphere Distributed Resource Scheduler (DRS) should place VMs on hosts in a resource pool. VM-Host affinity rules can be useful when host-based licensing requires VMs that are running certain applications to be placed on hosts that are licensed to run those applications. They can also be useful when virtual machines with workload-specific configurations require placement on hosts that have certain characteristics. The technical white paper *Best Practices for Performance Tuning of Telco and NFV Workloads in vSphere* (http://www.vmware.com/files/pdf/techpaper/vmware-tuning-telco-nfv-workloads-vsphere.pdf) provides several examples of virtual machine configurations that require specific host properties.

Starting with vCloud Director 9.5, service providers can expose VM groups to tenants by using VDC compute policies.

Affinity Rule Polarity

VMware Cloud Director VM-Host affinity rules are user-specified constraints that the system considers during the deployment process. There are two types, or polarities, of VM-Host affinity rules:

- A rule with a `Polarity` value of `Affinity` specifies that virtual machines in a VM group should be deployed on a host in a host group.
- A rule with a `Polarity` value of `Anti-Affinity` specifies that virtual machines in a VM group should not be deployed on a host in a host group.

You can make an affinity rule mandatory to specify that VM deployment must fail unless the specified placement objective can be achieved. When a virtual machine and host are the subject of a mandatory affinity rule, placement requirements dictated by the rule override any placement changes that might be initiated by vSphere services such as high availability and Storage DRS.

Host Groups and VM Groups

A vSphere VM-Host affinity rule is a rule of type **Virtual Machines to Hosts**, and must specify a host group and a VM group. Before a VMware Cloud Director system administrator can create a VM-Host affinity rule, a vSphere administrator must create at least one host DRS group in a resource pool mapped to a VMware Cloud Director Provider VDC, and a vSphere administrator or VMware Cloud Director system administrator must create a VM group in the same resource pool. VM-Host affinity rules express an affinity in all members of a VM group for all hosts in a host DRS group, so all hosts in a group should share one or more characteristics that a VM can require from a host. For example, you can group hosts on the basis of the application licenses they carry, and group VMs on the basis of the application licenses they require. You can then create VM-Host affinity rules that place VMs on hosts that carry the required licenses.

Because VM-Host affinity rules are properties of a resource pool, all members of groups that are subject to a rule must be deployed in the same resource pool. If a virtual machine or host is removed from the resource pool, the system removes it from any host group or VM group of which it is a member. The system does not update the group when the host or VM is returned to the resource pool.

Affinity Rule Interactions and Conflicts

All VM-Host affinity rules in a resource pool have the same precedence. This configuration has implications for how the rules interact. For example, a virtual machine that is a member of two VM groups, each of which is named in a different required VM-Host rule, can run only on hosts that belong to both of those host groups. When you create a VM-Host affinity rule, the system does not check for potential interactions of this kind.

The system does check for conflicts that could arise when applying multiple mandatory rules. For example, if you group VMs and hosts in a way that enables you to create a mandatory anti-affinity rule that applies to a VM and a host that are members of other groups that are subject to a different mandatory affinity rule, the system cannot apply to either rule. When two or more VM-Host affinity rules conflict in this way, the system applies the oldest rule and disables the others. You can correct the problem by making the rules optional, or by grouping the VMs and hosts in ways that minimize the chances of this sort of mandatory rule conflict occurring.

Affinity Rules and vSphere Resource Management

vSphere resource management features such as DRS, vSphere HA, and vSphere DPM never take any action that can violate a mandatory VM-Host affinity rule.

- DRS does not evacuate virtual machines to place a host in maintenance mode.
- DRS does not place virtual machines for power-on or load balance virtual machines.
- vSphere HA does not perform failovers.
- vSphere DPM does not optimize power management by placing hosts into standby mode.

To avoid these situations, be careful when you create more than one mandatory affinity rule that affects a specific VM-host pair. Be sure that the resource pool contains enough hosts so that losing a host does not leave the system with no host on which a VM that is governed by a rule can run. Rules that are not mandatory can be violated to allow the proper functioning of DRS, vSphere HA, and vSphere DPM.

Create or Update a Host Group

A host group is a vSphere host DRS group. A vSphere administrator must create host DRS groups in a resource pool mapped to a VMware Cloud Director Provider VDC before they can be used in VMware Cloud Director VM-Host affinity rules.

vSphere host DRS groups created in resource pools that are mapped to a Provider VDC appear in those resource pools and can be named in VM-Host affinity rules. For more information about host DRS groups, see the *VMware vSphere ESXi and vCenter Server Documentation*.

Procedure

- Host groups are properties of a resource pool.

  To retrieve the list of host groups in a resource pool, use a request like the one shown in Host Groups in a Resource Pool

Example: Host Groups in a Resource Pool

The `VMWProviderVdcResourcePoolSet` element of a `VMWProviderVdc` response includes a link of this form.

```
<vcloud:Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/id/hostGroups"
         type="application/vnd.vmware.admin.vmwHostGroupsType+xml" />
```

You can make a GET request to this link to list all vSphere host DRS groups created in that resource pool.

Request:

```
GET https://vcloud.example.com/api/admin/extension/resourcePool/83/hostGroups
```

This response shows a single host DRS group that includes two hosts.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWHostGroups
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <vmext:HostGroup>
      <vmext:HostGroupName>HostGroup-MSSQL</vmext:HostGroupName>
      <vmext:Hosts type="application/vnd.vmware.admin.vmwHostReferences+xml">
         <vmext:HostReference
            href="https://vcloud.example.com/api/admin/extension/host/95"
            name="ESXi37"
            type="application/vnd.vmware.admin.host+xml" />
         <vmext:HostReference
            href="https://vcloud.example.com/api/admin/extension/host/28"
            name="ESXi39"
            type="application/vnd.vmware.admin.host+xml" />
      </vmext:Hosts>
   </vmext:HostGroup>
</vmext:VMWHostGroups>
```

Create or Update a VM Group

A VM group is a collection of virtual machines with similar host requirements. The virtual machines must all be in the same resource pool.

Prerequisites

You must be a system administrator to create or update a VM group.

Procedure

1 VM groups are properties of a resource pool.

Each `VMWProviderVdcResourcePool` element in a `VMWProviderVdcResourcePoolSet` response includes links that you can use to create or update VM groups in the resource pool.

```
<?xml version="1.0" encoding="UTF-8"?>
<VMWProviderVdcResourcePoolSet ...>
   ...
   <VMWProviderVdcResourcePool
      primary="true">
      <Link
         rel="migrateVms"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/83/action/
migrateVms" />
      <Link
         rel="resourcePoolVmList"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/83/vmList" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/83/hostGroups"
         type="application/vnd.vmware.admin.vmwHostGroupsType+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/83/vmGroups"
         type="application/vnd.vmware.admin.vmwVmGroupType+xml" />
      <Link
         rel="add"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/83/vmGroups"
         type="application/vnd.vmware.admin.vmwVmGroupType+xml" />
      <Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/83/rules"
         type="application/vnd.vmware.admin.vmwVmHostAffinityRules+xml" />
      ...
   </VMWProviderVdcResourcePool>
   ...
</VMWProviderVdcResourcePoolSet>
```

2 To create a VM group in the resource pool, make a POST request to the `add` link for `vmGroups`.

The request body is a `VMWVmGroup` element. See Create a VM Group.

Example: Create a VM Group

The request body specifies a name for the group.

Request:

```
POST https://vcloud.example.com/api/admin/extension/resourcePool/83/vmGroups
Content-type: application/vnd.vmware.admin.vmwVmGroupType+xml
...
<VMWVmGroup
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud_v1.5="http://www.vmware.com/vcloud/v1.5">
   <VmGroupName>ExampleGroup</VmGroupName>
</VMWVmGroup>
```

The response is a Task. When the task completes, you can retrieve the `vmGroups` in the resource pool to see the new VM Group. The system supplies a unique identifier and a count of VMs, initially 0, in the group.

```
GET https://vcloud.example.com/api/admin/extension/resourcePool/83/vmGroups
...
<vmext:VMWVmGroups ...>
   <vmext:VmGroup
      href="https://vcloud.example.com/api/admin/extension/vmGroup/34">
      <vcloud:Link
         rel="remove"
         href="https://vcloud.example.com/api/admin/extension/vmGroup/34" />
      <vcloud:Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/vmGroup/34/vmsList"
         type="application/vnd.vmware.vcloud.VmGroupVmsRecord+xml" />
      <vcloud:Link
         rel="addVms"
         href="https://vcloud.example.com/api/admin/extension/vmGroup/name/foo/action/addVms"
         type="application/vnd.vmware.vcloud.vms+xml" />
      <vcloud:Link
         rel="removeVms"
         href="https://vcloud.example.com/api/admin/extension/vmGroup/name/foo/action/
removeVms"
         type="application/vnd.vmware.vcloud.vms+xml" />
      <vmext:VmGroupId>34</vmext:VmGroupId>
      <vmext:VmGroupName>ExampleGroup</vmext:VmGroupName>
      <vmext:vmCount>0</vmext:vmCount>
   </vmext:VmGroup>
   ...
</vmext:VMWVmGroups>
```

Create or Update a VM-Host Affinity Rule

A VM-Host affinity rule specifies a relationship between a host group and a VM group in the same resource pool. A system administrator can create, enable, disable or delete a VM-Host affinity rule.

After you create a VM-Host affinity rule, you can update it in the following ways:

- Enable the rule.

- Disable the rule.
- Delete the rule.

To make any other change (for example, to change the VM Group or Host Group), you must create a new rule.

vSphere VM-Host affinity rules that are created in a resource pool that is mapped to a Provider VDC are listed in the `VMWVmHostAffinityRules` element of the `VMWProviderVdc` response. For more information about host DRS VM-Host affinity, see the *VMware vSphere ESXi and vCenter Server Documentation*.

Prerequisites

- This operation is restricted to system administrators.
- You cannot create VM-Host affinity rule in a resource pool that does not contain at least one host group and one VM group.

Procedure

1 Choose a resource pool to contain the rule.

  Each `VMWProviderVdcResourcePool` element in a `VMWProviderVdcResourcePoolSet` response contains a link that you can use to retrieve or update the set of VM-Host affinity rules in the resource pool.

```
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWProviderVdcResourcePoolSet
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   ...
   <vmext:VMWProviderVdcResourcePool
      primary="true">
      ...
      <vcloud:Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/6/hostGroups"
         type="application/vnd.vmware.admin.vmwHostGroupsType+xml" />
      <vcloud:Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/6/vmGroups"
         type="application/vnd.vmware.admin.vmwVmGroupType+xml" />
      <vcloud:Link
         rel="add"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/6/vmGroups"
         type="application/vnd.vmware.admin.vmwVmGroupType+xml" />
      <vcloud:Link
         rel="add"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/6/rules"
         type="application/vnd.vmware.admin.vmwVmHostAffinityRules+xml" />
      <vcloud:Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/resourcePool/6/rules"
         type="application/vnd.vmware.admin.vmwVmHostAffinityRules+xml" />
      ...
   </vmext:VMWProviderVdcResourcePool>
   ...
</vmext:VMWProviderVdcResourcePoolSet>
```

To retrieve the list of VM-Host affinity rules in a resource pool, use a request like the one shown in Create a VM-Host Affinity Rule

2 Create a `VMWVmHostAffinityRule` element and POST it to the `add` link for rules in the resource pool.

   See Create a VM-Host Affinity Rule.

Example: Create a VM-Host Affinity Rule

This example creates and enables a rule that binds the Host Group shown in Host Groups in a Resource Pool to the VM Group created in Create a VM Group.

Request:

```
POST https://vcloud.example.com/api/admin/extension/resourcePool/6/rules
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWVmHostAffinityRule
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.vmwVmHostAffinityRule+xml">
   <vcloud:Name>MSSQL-VMs</vcloud:Name>
   <vcloud:IsEnabled>true</vcloud:IsEnabled>
   <vcloud:IsMandatory>true</vcloud:IsMandatory>
   <vcloud:Polarity>Affinity</vcloud:Polarity>
   <vmext:HostGroupName>HostGroup-MSSQL</vmext:HostGroupName>
   <vmext:VmGroupName>ExampleGroup</vmext:VmGroupName>
</vmext:VMWVmHostAffinityRule>
```

The response is a `Task`. When the task completes, you can make a request like this one to see the new rule and others in this resource pool. Each rule contains links you can use to enable, disable, or delete the rule.

```
GET https://vcloud.example.com/api/admin/extension/resourcePool/6/rules
...
<vmext:VMWVmHostAffinityRules ...>
   <vcloud:Link
      rel="add"
      href="https://vcloud.example.com/api/admin/extension/resourcePool/6/rules"
      type="application/vnd.vmware.admin.vmwVmHostAffinityRule+xml" />
   <vmext:VmHostAffinityRule
      id="85"
      href="https://vcloud.example.com/api/admin/extension/rule/85"
      type="application/vnd.vmware.admin.vmwVmHostAffinityRule+xml">
      <vcloud:Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/hostGroup/5"
         type="application/vnd.vmware.admin.vmwHostGroupType+xml" />
      <vcloud:Link
         rel="down"
         href="https://vcloud.example.com/api/admin/extension/vmGroup/34"
         type="application/vnd.vmware.admin.vmwVmGroupType+xml" />
      <vcloud:Link
         rel="disable"
         href="https://vcloud.example.com/api/admin/extension/rule/85/action/disable" />
      <vcloud:Link
         rel="remove"
         href="https://vcloud.example.com/api/admin/extension/rule/85" />
      vcloud:Name>MSSQL-VMs</vcloud:Name>
      <vcloud:IsEnabled>true</vcloud:IsEnabled>
      <vcloud:IsMandatory>true</vcloud:IsMandatory>
      <vcloud:Polarity>Affinity</vcloud:Polarity>
      <vmext:HostGroupName>HostGroup-MSSQL</vmext:HostGroupName>
      <vmext:VmGroupName>ExampleGroup</vmext:VmGroupName>
   </vmext:VmHostAffinityRule>
      ...
   <vmext:VmHostAffinityRule>
   </vmext:VmHostAffinityRule>
      ...
   <vmext:VmHostAffinityRule>
</vmext:VMWVmHostAffinityRules>
```

View and Modify Organization VDC Compute Policies

System administrators can create and publish VDC compute policies to organization VDCs. With the VMware Cloud Director API, you can view and modify organization VDC compute policies and change the default compute policies.

To create and manage provider and global VDC compute policies, you must use the VMware Cloud Director OpenAPI. See *Getting Started with VMware Cloud Director OpenAPI* at https://code.vmware.com.

Prerequisites

- To view VDC compute policies, you must be a **system administrator**, **organization administrator**, **vApp Author**, or **vApp User**.
- To update the VDC compute policies, you must be a **system administrator**.
- To change the default VDC compute policy, you must be a **system administrator** or **organization administrator**.

Procedure

1 View and update the compute policies that are published to an organization VDC.

   a To retrieve the compute policies of an organization VDC, use a GET request to the `computePolicies` link of the target VDC.

   ```
   GET https://vcloud.example.com/api/vdc/83/computePolicies
   ```

   The response contains a `VdcComputePolicyReference` element for each VDC compute policy.

   b To update the compute policies of an organization VDC, modify the `VdcComputePolicyReference` element and use a PUT request to the `computePolicies` link of the target VDC.

   ```
   PUT https://vcloud.example.com/api/admin/vdc/83/computePolicies
   ...
   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
   <VdcComputePolicyReferences
      xmlns="http://www.vmware.com/vcloud/v1.5"
      ...>
      <VdcComputePolicyReference
         href=""
         id="57"
         name="fast_mssql_updated"
         type="application/json"/>
   </VdcComputePolicyReferences>
   ```

2 View and update the default compute policy of an organization VDC.

   a To retrieve the default compute policy of an organization VDC, retrieve an XML representation of the target VDC.

   ```
   GET https://vcloud.example.com/api/vdc/83
   ```

   The response contains a `DefaultComputePolicy` element with the UUID and name of the default compute policy.

   b To change the default compute policy of an organization VDC, modify the `DefaultComputePolicy` element and use a PUT request to the target VDC.

   ```
   PUT https://vcloud.example.com/api/vdc/83
   ...
   ...
      <IsEnabled>true</IsEnabled>
      <VdcStorageProfiles>
         <VdcStorageProfile
            .../>
       </VdcStorageProfiles>
       <DefaultComputePolicy id="57" name="fast_mssql_updated" 
   type="com.vmware.vcloud.entity.vdcComputePolicy"/>
       <VCpuInMhz2>1000</VCpuInMhz2>
       <ResourceGuaranteedMemory>0.2</ResourceGuaranteedMemory>
       <ResourceGuaranteedCpu>0.2</ResourceGuaranteedCpu>
   ...
   ...
   ```

Change the Maximum VDC Compute Policy

To limit the maximum compute resources that tenants can allocate to individual virtual machines within an organization VDC, system administrators can change the maximum VDC compute policy.

Prerequisites

To change the maximum VDC compute policy, you must be a **system administrator** or **organization administrator**.

Procedure

1 To retrieve the max compute policy of an organization VDC, retrieve an XML representation of the target VDC.

   ```
   GET https://vcloud.example.com/api/vdc/83
   ```

   The response contains a `MaxComputePolicy` element with `href`.

   ```
   <MaxComputePolicy 
   href="https://vcloud.example.com/cloudapi/1.0.0/vdcs/83/maxComputePolicy" 
   type="application/json" />
   ```

2. To change the max compute policy of an organization VDC, modify the `MaxComputePolicy` element and use a PUT request to the target VDC.

   ```
     PUT https://vcloud.example.com/api/vdc/83
   ...
   ...
   <IsEnabled>true</IsEnabled>
   <VdcStorageProfiles>
   <VdcStorageProfile
   .../>
   </VdcStorageProfiles>
   <MaxComputePolicy id="59" name="flex_max_sizing_policy"
   type="com.vmware.vcloud.entity.vdcComputePolicy"/>
   <VCpuInMhz2>1000</VCpuInMhz2>
   <ResourceGuaranteedMemory>0.2</ResourceGuaranteedMemory>
   <ResourceGuaranteedCpu>0.2</ResourceGuaranteedCpu>
   ...
   ...
   ```

What to do next

From VMware Cloud Director API version 33.0, to view and update the maximum compute policy of an organization VDC, you can use the VMware Cloud Director OpenAPI. See *Getting Started with VMware Cloud Director OpenAPI* at https://code.vmware.com.

Creating and Managing VDC Templates

A VDC template specifies a configuration for an organization VDC and, optionally, an Edge Gateway and organization VDC network. System administrators who want to enable organization administrators to create these resources in their organization can create VDC templates and share them with those organizations.

By creating and sharing VDC templates, system administrator can enable self-service provisioning of organization VDCs while retaining administrative control over allocation of system resources such as Provider VDCs and external networks. Organization administrators, or any role that has rights to view and instantiate VDC templates, use an instantiation operation to create organization VDCs from templates. See Create a VDC from a Template.

Provider VDC Specification

Although you can create a VDC template that specifies a single Provider VDC to support all organization VDCs instantiated from that template, you can also let the system spread organization VDC load across multiple Provider VDCs if they are present in the system. A VDC template can include multiple `ProviderVdcReference` elements, each of which must be paired with an external network backed by the same vCenter server that backs the Provider VDC. Each time this sort of template is instantiated, the system selects a Provider VDC and its paired network to back the organization VDC that the instantiation creates. Selection criteria are intended to choose a Provider VDC that has the necessary CPU, memory, and storage resources to support a new organization VDC.

You cannot delete a Provider VDC, external network, storage profile, or network pool referenced in a VDC template until the VDC template and all organization VDCs created from it are removed from the system.

Note You can create organization virtual data center templates only for virtual data centers backed by NSX Data Center for vSphere.

Edge Gateway and Routed Network Specification

A `VMWVdcTemplate` element can specify the configuration of an Edge Gateway and an organization VDC network. These objects are created in the VDC when the template is instantiated, and an organization administrator can update them by using the workflows described in Network Administration. An organization VDC created from a template that does not include an Edge Gateway can contain only isolated networks.

Allocation Models

Every VDC template includes a `VdcTemplateSpecification` element that specifies a set of properties similar to those specified in the `CreateVdcParams` element that is used to create an organization VDC. See Create an Organization VDC with Pay As You Go Reservation Allocation Model. The subtypes of `VdcTemplateSpecificationType` specify one of the allocation models supported for organization VDCs. See Create a VDC Template.

Create a VDC Template

To create a VDC template, POST a `VMWVdcTemplate` element to the system URL for adding VDC templates.

In its simplest form, a VDC template must include references to the following objects:

- A Provider VDC
- A Provider VDC storage profile defined in that Provider VDC

When this form of VDC template is instantiated in an organization, it creates an organization VDC that has no Edge Gateway or routed networking capability. An organization administrator can create isolated networks in it. A system administrator can add networking capability to an organization VDC, whether or not it was created from a template, by following the procedures documented in Create an Edge Gateway and Create an Organization VDC Network.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the vSphere platform extensions.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension
```

2 Examine the response to locate the `Link` element that contains the URL for adding VDC templates to the system.

This element has a `rel` attribute value of `add` and a `type` attribute value of `application/vnd.vmware.admin.vmwVdcTemplate+xml`, as shown here:

```
<Link
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/action/vdcTemplates"
   type="application/vnd.vmware.admin.vmwVdcTemplate+xml"/>
```

3 Choose a provider VDC to supply resources for organization VDCs that are created by instantiating this template.

You can use a query like this one to list all the Provider VDCs in the system.

```
GET https://vcloud.example.com/api/query?type=providerVdc&format=references
```

4 Create a `VMWVdcTemplate` request body.

   a Include a `ProviderVdcReference` that references the Provider VDC that you chose in Step 3.

   b Include a `VdcTemplateSpecification` element that specifies how compute resources are allocated by organization VDCs created from this template.

   Each resource allocation model is defined in a subtype of `VMWVdcTemplateSpecificationType`. Specify one of the following allocation models by setting the value of the `VdcTemplateSpecification` element's `type` attribute as shown.

   AllocationVapp

```
<VdcTemplateSpecification
      type="VMWAllocationVappVdcTemplateSpecificationType">
```

Specifies an `AllocationVApp` (Pay as you go) allocation model. Resources are committed to the organization VDC only when vApps are created in it. When you use this allocation model, `Limit` values that you specify for `Memory` and `CPU` are ignored when you create a vApp, and are returned as `0` when you retrieve a vApp. Resources available to this kind of organization VDC can grow or shrink as needed when its provider VDC has multiple resource pools.

   AllocationPool

```
<VdcTemplateSpecification
      type="VMWAllocationPoolVdcTemplateSpecificationType">
```

Specifies an `AllocationPool` allocation model. Only a percentage of the resources you allocate are committed to the organization VDC.

   ReservationPool

```
<VdcTemplateSpecification
      type="VMWReservationPoolVdcTemplateSpecificationType">
```

Specifies a `ReservationPool` allocation model. All the resources you allocate are committed as a pool to the organization VDC. vApps in VDCs that support this allocation model can specify values for resources and limitations.

   c Include at least one `VdcStorageProfile` element that specifies a `ProviderVdcStorageProfile` defined in the Provider VDC you chose in Step 3.

   You can use a query like this one to list all the Provider VDC storage profiles in the system. A filter expression constrains the output to just those Provider VDC storage profiles that are associated with a single Provider VDC, **`vCenter01`**, in this case.

```
GET https://vcloud.example.com/api/query?
type=providerVdc&format=references&filter=providerVdc==vCenter01
```

The specified Provider VDC storage profiles are made available in organization VDCs created from this template.

5. POST the `VMWVdcTemplate` request body to the system's `add` link for `vdcTemplates`.

   See the request portion of Create a VDC Template.

Results

The system creates the VDC template and returns a `VMWVdcTemplate` element that includes a set of `Link` elements that you can use to access, remove, or modify the new template.

Example: Create a VDC Template

This request creates a VDC template that, when instantiated, adds an `AllocationVapp` VDC to an organization. An organization VDC created by instantiating this template has many of the same properties as the one created in Create an Organization VDC with Pay As You Go Reservation Allocation Model.

Request:

```
POST https://vcloud.example.com/api/admin/extension/vdcTemplates
Content-Type: application/vnd.vmware.admin.vmwVdcTemplate+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWVdcTemplate
   ...
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   name="example-vdc-template">
   <Description>Example AllocationVapp VDC Template without Edge Gateway</Description>
   <vmext:TenantName>PayAsYouGo-VDCTemplate-no-net</vmext:TenantName>
   <vmext:TenantDescription>PayAsYouGo VDC with no networking</vmext:TenantDescription>
   <vmext:ProviderVdcReference
      href="https://vcloud.example.com/api/admin/providervdc/35"
      name="vCenter01"
      type="application/vnd.vmware.admin.providervdc+xml" />
   <vmext:VdcTemplateSpecification
      xsi:type="vmext:VMWAllocationVappVdcTemplateSpecificationType">
      <NicQuota>100</NicQuota>
      <VmQuota>50</VmQuota>
      <ProvisionedNetworkQuota>100</ProvisionedNetworkQuota>
      <StorageProfile name="Bronze">
         <Enabled>true</Enabled>
         <Units>MB</Units>
         <Limit>2097152</Limit>
         <Default>true</Default>
      </StorageProfile>
      <vmext:ThinProvision>false</vmext:ThinProvision>
      <vmext:FastProvisioningEnabled>false</vmext:FastProvisioningEnabled>
      <CpuAllocationMhz>2048</CpuAllocationMhz>
      <CpuLimitMhzPerVcpu>1000</CpuLimitMhzPerVcpu>
      <MemoryAllocationMB>2048</MemoryAllocationMB>
      <CpuGuaranteedPercentage>1</CpuGuaranteedPercentage>
      <MemoryGuaranteedPercentage>1</MemoryGuaranteedPercentage>
   </vmext:VdcTemplateSpecification>
</vmext:VMWVdcTemplate>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.vmwVdcTemplate+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWVdcTemplate
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   name="example-vdc-template" ...>
   <vcloud:Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/extension/vdcTemplate/100"
      type="application/vnd.vmware.admin.vmwVdcTemplate+xml"/>
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/vdcTemplate/100"/>
   <vcloud:Link
      rel="down"
      href="https://vcloud.example.com/api/admin/extension/vdcTemplate/100/controlAccess/"
      type="application/vnd.vmware.vcloud.controlAccess+xml"/>
   <vcloud:Link
      rel="controlAccess"
      href="https://vcloud.example.com/api/admin/extension/vdcTemplate/100/action/
controlAccess"
      type="application/vnd.vmware.vcloud.controlAccess+xml"/>
   <vcloud:Link
      rel="alternate"
      href="https://vcloud.example.com/api/vdcTemplate/100"
      type="application/vnd.vmware.admin.vdcTemplate+xml"/>
   <vcloud:Description>Example AllocationVapp VDC Template without Edge Gateway</
vcloud:Description>
   <vmext:TenantName>PayAsYouGo-VDCTemplate-no-net</vmext:TenantName>
   <vmext:TenantDescription>PayAsYouGo VDC with no networking</vmext:TenantDescription>
   <vmext:ProviderVdcReference
      href="https://vcloud.example.com/api/admin/providervdc/35"
      name="pvdc"
      type="application/vnd.vmware.admin.providervdc+xml"/>
   <vmext:VdcTemplateSpecification
      xsi:type="vmext:VMWAllocationVappVdcTemplateSpecificationType">
      ...
   </vmext:VdcTemplateSpecification>
</vmext:VMWVdcTemplate>
```

Create a VDC Template That Includes Routed Networking

A VDC template can specify configurations for an Edge Gateway and organization VDC network. Each time this kind of template is instantiated, the resulting organization VDC contains an Edge

Gateway and routed network. An organization administrator can then configure Edge Gateway services and add routed or isolated organization VDC networks as needed.

A VDC template with routed networking must include references to the following objects:

- A Provider VDC
- A Provider VDC storage profile defined in that Provider VDC
- An external network available in that Provider VDC
- A network pool associated with that Provider VDC

When it is instantiated, this form of VDC template creates an organization VDC that includes an Edge Gateway with a single uplink interface to the specified external network, and a single routed organization VDC network.

Prerequisites

This operation is restricted to system administrators.

Create a `VMWVdcTemplate` request body. The procedure shown here adds routed networking capability to a template like the one created in Create a VDC Template.

Procedure

1 List the external networks and network pools associated with the Provider VDC referenced in the template's `ProviderVdcReference` element.

   Use a request like this one to retrieve the XML representation of the Provider VDC:

   ```
   GET https://vcloud.example.com/api/admin/extension/providervdc/35
   ```

   The `AvailableNetworks` element in the response lists the external networks associated with that Provider VDC. The `NetworkPoolReferences` element in the response lists references to all network pools associated with that Provider VDC.

   ```
   <vmext:VMWProviderVdc ... >
      ...
      <vcloud:AvailableNetworks>
         <vcloud:Network
            type="application/vnd.vmware.admin.network+xml"
            name="VC1-VLAN48"
            href="https://vcloud.example.com/api/admin/network/297" />
         <vcloud:Network ... />
         <vcloud:Network ... />
      </vcloud:AvailableNetworks>
      ...
      <vcloud:NetworkPoolReferences>
         <vcloud:NetworkPoolReference
            type="application/vnd.vmware.admin.networkPool+xml"
   ```

```
          name="VC1-VXLAN"
          href="https://vcloud.example.com/api/admin/extension/networkPool/313" />
   </vcloud:NetworkPoolReferences>
</vmext:VMWProviderVdc>
```

.

2 Add a `GatewayConfiguration` element to the `VdcTemplateSpecification`.

   This `GatewayConfiguration` must include exactly one `GatewayInterface`. Its `Network` element must reference an external network from the list you retrieved in Step 1 and its `InterfaceType` must have a value of `uplink`. You cannot specify a `BackwardCompatibilityMode` or any `SubnetParticipation` in this `GatewayConfiguration`.

   a Include a `Network` element that configures the initial organization VDC network for organization VDCs created from this template

      This `Network` must specify a `FenceMode` of `natRouted`, and cannot include any of the following elements:

      - `BackwardCompatibilityMode`
      - `EdgeGateway`
      - `NetworkFeatures`
      - `RouterInfo`
      - `ServiceConfig`
      - `SyslogServerSettings`

      Its `IpScopes` element must contain exactly one `IpScope`, which must have an `IsInherited` value of `false` and an `IsEnabled` value of `true`. This `IpScope` cannot include any of the following elements:

      - AllocatedIpAddresses
      - Dns1
      - Dns2
      - DnsSuffix
      - SubAllocations

      See the request portion of Create a VDC Template That Includes Routed Networking.

3 Specify a network pool for the organization VDCs created from this template to use.

   You can add a `NetworkPoolReference` element from the list you retrieved in Step 1 or you can add an empty `AutomaticNetworkPoolReference` element to specify that organization VDCs created from this template will use the automatically created VXLAN pool associated with the Provider VDC specified in this template. See the request portion of Create a VDC Template That Includes Routed Networking.

4 POST the `VMWVdcTemplate` request body to the system's `add` link for `vdcTemplates`.

See the request portion of Create a VDC Template That Includes Routed Networking.

Results

The system creates the new VDC template and returns a `VMWVdcTemplate` element that includes a set of `Link` elements that you can use to access, remove, or modify the new template.

Example: Create a VDC Template That Includes Routed Networking

This example extends the one in Create a VDC Template to create a VDC template that, when instantiated, adds a VDC that contains an Edge Gateway and a routed organization VDC network to the organization. An organization VDC that is created by instantiating this template has properties similar to the one created in Create an Organization VDC with Pay As You Go Reservation Allocation Model. The Edge Gateway and organization VDC network it contains are similar to the ones created in Create an Edge Gateway and Create an Organization VDC Network With a Routed Connection.

Request:

```
POST https://vcloud.example.com/api/admin/extension/vdcTemplates
Content-Type: application/vnd.vmware.admin.vmwVdcTemplate+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWVdcTemplate
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   name="example-vdc-template">
   <Description>Example AllocationVapp VDC Template with Gateway</Description>
   <vmext:TenantName>PayAsYouGo-VDCTemplate</vmext:TenantName>
   <vmext:TenantDescription>PayAsYouGo-VdcTemplate</vmext:TenantDescription>
   <vmext:ProviderVdcReference
      href="https://vcloud.example.com/api/admin/providervdc/35"
      name="vCenter01"
      type="application/vnd.vmware.admin.providervdc+xml"/>
   <vmext:VdcTemplateSpecification
      xsi:type="vmext:VMWAllocationVappVdcTemplateSpecificationType">
      <NicQuota>100</NicQuota>
      <VmQuota>50</VmQuota>
      <ProvisionedNetworkQuota>100</ProvisionedNetworkQuota>
      <GatewayConfiguration>
         <Gateway name="theEdge">
            <Description>Edge Gateway defined by VDC template</Description>
            <Configuration>
               <GatewayBackingConfig>compact</GatewayBackingConfig>
               <GatewayInterfaces>
                  <GatewayInterface>
                     <Name>uplink1</Name>
                     <DisplayName>Uplink interface defined by VDC template</DisplayName>
                     <Network href="https://vcloud.example.com/api/admin/network/297"/>
                     <InterfaceType>uplink</InterfaceType>
                  </GatewayInterface>
               </GatewayInterfaces>
               <HaEnabled>false</HaEnabled>
               <UseDefaultRouteForDnsRelay>false</UseDefaultRouteForDnsRelay>
            </Configuration>
         </Gateway>
         <Network name="RoutedOVDCNet">
            <Description>Routed through an Edge Gateway</Description>
            <Configuration>
               <IpScopes>
                  <IpScope>
                     <IsInherited>false</IsInherited>
                     <Gateway>192.168.0.1</Gateway>
                     <Netmask>255.255.255.0</Netmask>
                     <IpRanges>
                        <IpRange>
                           <StartAddress>192.168.0.100</StartAddress>
                           <EndAddress>192.168.0.199</EndAddress>
                        </IpRange>
                     </IpRanges>
                  </IpScope>
               </IpScopes>
               <FenceMode>natRouted</FenceMode>
            </Configuration>
            <IsShared>false</IsShared>
         </Network>
      </GatewayConfiguration>
      <StorageProfile name="Bronze">
         <Enabled>true</Enabled>
         <Units>MB</Units>
         <Limit>2097152</Limit>
         <Default>true</Default>
      </StorageProfile>
      <vmext:ThinProvision>false</vmext:ThinProvision>
      <vmext:FastProvisioningEnabled>false</vmext:FastProvisioningEnabled>
      <vmext:NetworkPoolReference
         href="https://vcloud.example.com/api/admin/extension/networkPool/313"/>
      <CpuAllocationMhz>2048</CpuAllocationMhz>
      <CpuLimitMhzPerVcpu>1000</CpuLimitMhzPerVcpu>
      <MemoryAllocationMB>2048</MemoryAllocationMB>
      <CpuGuaranteedPercentage>1</CpuGuaranteedPercentage>
      <MemoryGuaranteedPercentage>1</MemoryGuaranteedPercentage>
   </vmext:VdcTemplateSpecification>
</vmext:VMWVdcTemplate>
```

The response is similar to the one shown in Create a VDC Template.

Create a VDC Template That Includes Routed Networking and Multiple Provider VDCs

If the system includes more than one Provider VDC, and at least two of those Provider VDCs use a common set of storage profiles, a `VMWVdcTemplate` in that system can include multiple `ProviderVdcReference` elements. Each time this sort of template is instantiated, the system evaluates the current resource commitments of each of the referenced Provider VDCs and chooses the one best able to support a new organization VDC.

A VDC template that includes routed networking and specifies multiple Provider VDCs must include references to the following objects:

- Two or more candidate Provider VDCs that share at least one storage profile. Each of the Provider VDCs must have at least one available external network.
- A reference to a Provider VDC storage profile defined in all candidate Provider VDCs
- A network pool specification.

When this form of VDC template is instantiated, it creates an organization VDC that is backed by one of the candidate Provider VDCs.

Note When you specify multiple Provider VDCs in a template that includes networking, you must also add a `Binding` element to each `ProviderVdcReference`. A `Binding` associates a symbolic name with an external network in the Provider VDC. You use this symbolic name as the value of the `href` attribute of the `Network` configured in the template. When the template is instantiated, the system replaces the symbolic name with a reference to the network specified in the `Value` element of the `Binding`.

Prerequisites

This operation is restricted to system administrators.

Use a procedure like the one shown in Create a VDC Template or Create a VDC Template That Includes Routed Networking to create a `VMWVdcTemplate` request body. You can specify multiple Provider VDCs in a VDC template whether or not the template includes routed networking. The procedure shown here adds another Provider VDC to the template shown in Create a VDC Template That Includes Routed Networking.

Procedure

1 List the Provider VDCs that exist in the system.

You can use a query like this one to list all the Provider VDCs in the system.

```
GET https://vcloud.example.com/api/query?type=providerVdc&format=references
```

2 List the external networks and network pools available in each Provider VDCto specify in this VDC template.

Use a request like this one to retrieve the XML representation of the Provider VDC:

```
GET https://vcloud.example.com/api/admin/extension/providervdc/35
```

The AvailableNetworks element in the response lists the external networks created in that Provider VDC. The NetworkPoolReferences element in the response lists references to all network pools created in that Provider VDC.

```
<vmext:VMWProviderVdc ... >
   ...
   <vcloud:AvailableNetworks>
      <vcloud:Network
         type="application/vnd.vmware.admin.network+xml"
         name="VC1-VLAN48"
         href="https://vcloud.example.com/api/admin/network/297" />
      <vcloud:Network ... />
      <vcloud:Network ... />
   </vcloud:AvailableNetworks>
   ...
   <vcloud:NetworkPoolReferences>
      <vcloud:NetworkPoolReference
         type="application/vnd.vmware.admin.networkPool+xml"
         name="VC1-VXLAN"
         href="https://vcloud.example.com/api/admin/extension/networkPool/313" />
   </vcloud:NetworkPoolReferences>
</vmext:VMWProviderVdc>
```

.

3 For each Provider VDC that you include in this template, create a ProviderVdcReference that references the Provider VDC and includes a Binding element that references an external network available in that Provider VDC.

4 POST the VMWVdcTemplate request body to the system's add link for vdcTemplates.

   See the request portion of Create a VDC Template With Routed Networking and Multiple Provider VDCs.

Results

The system creates the new VDC template and returns a VMWVdcTemplate element that includes a set of Link elements that you can use to access, remove, or modify the new template.

Example: Create a VDC Template With Routed Networking and Multiple Provider VDCs

This request creates a VDC template that is identical to the one shown in Create a VDC Template That Includes Routed Networking, but adds a second ProviderVdcReference and includes Binding elements that provide symbolic references to external networks.

The Binding element in each ProviderVdcReference pairs a Name element that contains a user-specified identifier in URN format with a Value element that specifies the href of an external network in the candidate Provider VDC. The GatewayConfiguration of the template uses that URN instead of a URL for the href attribute of its Network element.

```
<Network href="urn:vcloud:binding:35c80ae2-36ef-4e2f-a430-ca704db9709f"/>
```

When the template is instantiated, this URN is replaced by the network URL in the `Value` part of the `Binding` associated with the `ProviderVdcReference` that the system selects during instantiation.

This example also uses the `AutomaticNetworkPoolReference` flag to specify that organization VDCs created by instantiating this template use the VXLAN network pool that was created automatically when the Provider VDC was created. If you decide not to use `AutomaticNetworkPoolReference` in a VDC template that specifies multiple Provider VDCs, you must specify a network pool that exists in one of the candidate Provider VDCs. All organization VDCs that are created from this template use that pool, regardless of the Provider VDC that is chosen during instantiation.

Request:

```
POST https://vcloud.example.com/api/admin/extension/vdcTemplates
Content-Type: application/vnd.vmware.admin.vmwVdcTemplate+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VMWVdcTemplate
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   name="example-vdc-template-2PVDCs">
   <Description>Example AllocationVapp VDC Template with Gateway</Description>
   <vmext:TenantName>PayAsYouGo-VDCTemplate</vmext:TenantName>
   <vmext:TenantDescription>PayAsYouGo-VdcTemplate</vmext:TenantDescription>
   <vmext:ProviderVdcReference
      href="https://vcloud.example.com/api/admin/providervdc/35"
      name="vCenter01"
      type="application/vnd.vmware.admin.providervdc+xml">
      <vmext:Binding>
         <vmext:Name>urn:vcloud:network:35c80ae2-36ef-4e2f-a430-ca704db9709f</vmext:Name>
         <vmext:Value href="https://vcloud.example.com/api/admin/network/297"></vmext:Value>
      </vmext:Binding>
   </vmext:ProviderVdcReference>
   <vmext:ProviderVdcReference
      href="https://vcloud.example.com/api/admin/providervdc/39"
      name="vCenter02"
      type="application/vnd.vmware.admin.providervdc+xml">
      <vmext:Binding>
         <vmext:Name>urn:vcloud:network:35c80ae2-36ef-4e2f-a430-ca704db9709f</vmext:Name>
         <vmext:Value href="https://vcloud.example.com/api/admin/network/99"></vmext:Value>
      </vmext:Binding>
   </vmext:ProviderVdcReference>
   <vmext:VdcTemplateSpecification
      xsi:type="vmext:VMWAllocationVappVdcTemplateSpecificationType">
      <NicQuota>100</NicQuota>
      <VmQuota>50</VmQuota>
      <ProvisionedNetworkQuota>100</ProvisionedNetworkQuota>
      <GatewayConfiguration>
         <Gateway name="theEdge">
            <Description>Edge Gateway defined by VDC template</Description>
            <Configuration>
```

```
               <GatewayBackingConfig>compact</GatewayBackingConfig>
               <GatewayInterfaces>
                  <GatewayInterface>
                     <Name>uplink1</Name>
                     <DisplayName>Uplink interface defined by VDC template</DisplayName>
                     <Network
                        href="urn:vcloud:binding:35c80ae2-36ef-4e2f-a430-ca704db9709f"/>
                     <InterfaceType>uplink</InterfaceType>
                  </GatewayInterface>
               </GatewayInterfaces>
               <HaEnabled>false</HaEnabled>
               <UseDefaultRouteForDnsRelay>false</UseDefaultRouteForDnsRelay>
            </Configuration>
         </Gateway>
         <Network name="RoutedOVDCNet">
            <Description>Routed through an Edge Gateway</Description>
            <Configuration>
               <IpScopes>
                  <IpScope>
                     <IsInherited>false</IsInherited>
                     <Gateway>192.168.0.1</Gateway>
                     <Netmask>255.255.255.0</Netmask>
                     <IpRanges>
                        <IpRange>
                           <StartAddress>192.168.0.100</StartAddress>
                           <EndAddress>192.168.0.199</EndAddress>
                        </IpRange>
                     </IpRanges>
                  </IpScope>
               </IpScopes>
               <FenceMode>natRouted</FenceMode>
            </Configuration>
            <IsShared>false</IsShared>
         </Network>
      </GatewayConfiguration>
      <StorageProfile name="Bronze">
         <Enabled>true</Enabled>
         <Units>MB</Units>
         <Limit>2097152</Limit>
         <Default>true</Default>
      </StorageProfile>
      <vmext:ThinProvision>false</vmext:ThinProvision>
      <vmext:FastProvisioningEnabled>false</vmext:FastProvisioningEnabled>
      <vmext:AutomaticNetworkPoolReference/>
      <CpuAllocationMhz>2048</CpuAllocationMhz>
      <CpuLimitMhzPerVcpu>1000</CpuLimitMhzPerVcpu>
      <MemoryAllocationMB>2048</MemoryAllocationMB>
      <CpuGuaranteedPercentage>1</CpuGuaranteedPercentage>
      <MemoryGuaranteedPercentage>1</MemoryGuaranteedPercentage>
   </vmext:VdcTemplateSpecification>
</vmext:VMWVdcTemplate>
```

Controlling Access to VDC Templates

Upon creation, a VDC template is not accessible to any organization in the system. To enable organizations to create VDCs from a template, a system administrator must use the VMware Cloud Director API access control mechanism to grant those organizations `ReadOnly` access to the template, and must also grant organization members rights to view and instantiate VDC templates.

A system administrator can make a request similar to the ones described in Controlling Access to vApps and Catalogs to control access to VDC templates.

Important The rights to view and instantiate VDC templates are restricted to the system administrator by default. The system administrator must explicitly grant those rights to organization members or roles to enable use of VDC templates in an organization.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the VDC template to find its `controlAccess` link.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/VdcTemplate/100
```

The `controlAccess` link has the following form:

```
<vcloud:Link
   rel="controlAccess"
   href="https://vcloud.example.com/api/admin/extension/vdcTemplate/100/action/
controlAccess"
   type="application/vnd.vmware.vcloud.controlAccess+xml"/>
```

2 Create a `ControlAccessParams` request body that grants `ReadOnly` access to the organizations that you want to have access to the template .

The only `AccessLevel` that you can grant for a `VdcTemplate` is `ReadOnly`. You cannot set `IsSharedToEveryone` to true.

3 POST the `ControlAccessParams` request body to the link you retrieved in Step 1.

Results

A link to the specified VDC template is returned in the `vdcTemplates` list of those organizations to which access was granted. See Create a VDC from a Template

Example: Grant Access to a VDC Template

This request grants access to a VDC template to two organizations.

Request:

```
POST https://vcloud.example.com/api/admin/extension/vdcTemplate/100/action/controlAccess
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<ControlAccessParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns="http://www.vmware.com/vcloud/v1.5">
   <IsSharedToEveryone>false</IsSharedToEveryone>
   <AccessSettings>
      <AccessSetting>
         <Subject href="https://vcloud.example.com/api/org/5"></Subject>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
      <AccessSetting>
         <Subject href="https://vcloud.example.com/api/org/26"></Subject>
         <AccessLevel>ReadOnly</AccessLevel>
      </AccessSetting>
   </AccessSettings>
</ControlAccessParams>
```

The response is a `ControlAccessParams` element.

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.controlAccess+xml
...
<ControlAccessParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns="http://www.vmware.com/vcloud/v1.5" \>
   ...
</ControlAccessParams>
```

Import a Virtual Machine from vCenter

A system administrator can import virtual machines from the inventory of any vCenter server registered to VMware Cloud Director. You can import the virtual machines to any VDC in a cloud, and you can import them in vApp or vApp template form.

When you import a virtual machine from vCenter, you must specify the following items:

- A target VDC to receive the import.
- A form for the imported virtual machine to take. Choose vApp or vApp template.
- Whether to remove the source virtual machine from vCenter inventory after the import is complete.

Prerequisites

- This operation is restricted to system administrators.

- Identify the virtual machine to import. See Retrieve a List of Virtual Machines from a vCenter Server.

Procedure

1 Choose whether to import the virtual machine as a vApp or vApp template.

  The `VimServer` element that represents the vCenter server from which you import the virtual machine contains two links that import virtual machines. One has the following form, and imports the virtual machine as a vApp.

```
<vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.importVmAsVAppParams+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/importVmAsVApp" />
```

  The other has the following form, and imports the virtual machine as a vApp template.

```
<vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.importVmAsVAppTemplateParams+xml"
      href="https://vcloud.example.com/api/admin/extension/vimServer/9/
importVmAsVAppTemplate" />
```

2 (Optional) If you plan to import the virtual machine as a vApp template, identify a catalog where you want to place a reference to the template.

Import a Virtual Machine as a vApp

To import a virtual machine as a vApp, a system administrator can make a request to the `importVmAsVApp` link of the `VimServer` that manages the virtual machine.

Prerequisites

- This operation is restricted to system administrators.
- Identify the virtual machine to import. See Retrieve a List of Virtual Machines from a vCenter Server.

Procedure

1 Create an `ImportVmAsVAppParams` element that specifies the `VmMoRef` of the source virtual machine and a target VDC to hold the imported vApp.

2 POST the `ImportVmAsVAppParams` element to the `importVmAsVApp` link of the source vCenter server.

Example: Import a Virtual Machine as a vApp

This example imports one of the virtual machines shown in the response portion of Retrieve a List of Virtual Machines from a vCenter Server. The request body is an `ImportVmAsVAppParams` element whose `sourceMove` attribute specifies that the source virtual machine should remain in vCenter inventory after the import is complete. The request body includes the `href` of the VDC that receives the import and a `VmMoRef` element that contains the managed object reference of the virtual machine to import. The response is an unresolved `vApp` body that contains a `Task` that tracks the import.

Request:

```
POST https://vcloud.example.com/api/admin/extension/vimServer/9/importVmAsVapp
Content-type: application/vnd.vmware.admin.importVmAsVAppParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ImportVmAsVAppParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   name="ImportedWin2K8"
   sourceMove="false">
   <VmMoRef>vm-43</VmMoRef>
   <Vdc
      href="http://vcloud.example.com/api/vdc/2" />
</ImportVmAsVAppParams>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vApp+xml
...
<VApp ...
   status="0"
   name="ImportedWin2K8"
   type="application/vnd.vmware.vcloud.vApp+xml"
   href="https://vcloud.example.com/api/vApp/vapp-102" ... >
   ...
   <Description />
   <Tasks>
      <Task
         operation=”Busy Virtual Application Win2K8 ”>
         ...
      </Task>
   </Tasks>
</VApp>
```

Import a Virtual Machine as a vApp Template

To import a virtual machine as a vApp template, a system administrator can make a request to the `importVmAsVAppTemplate` link of the `VimServer` that manages the virtual machine.

Prerequisites

- This operation is restricted to system administrators.
- Identify the virtual machine to import. See Retrieve a List of Virtual Machines from a vCenter Server.

Procedure

1 Create an `ImportVmAsVAppTemplateParams` element that specifies the `VmMoRef` of the source virtual machine, a target VDC to hold the imported vApp template, and an optional catalog where you want to place a reference to the template.

2 POST the `ImportVmAsVAppTemplateParams` element to the `importVmAsVAppTemplate` link of the source vCenter server.

Example: Import a Virtual Machine as a vApp Template

This example imports one of the virtual machines shown in the response portion of Retrieve a List of Virtual Machines from a vCenter Server as a vApp template. The request body is an `ImportVmAsVAppTemplateParams` element whose `sourceMove` attribute specifies that the source virtual machine should remain in vCenter inventory after the import is complete. The request body includes the `href` of the VDC that receives the import, a `VmMoRef` element that contains the managed object reference of the virtual machine to import, and a `Catalog` element that references the catalog to which the imported template should be added. The response is an unresolved `VAppTemplate` body that contains a `Task` that tracks the import.

Request:

```
POST https://vcloud.example.com/api/admin/extension/vimServer/9/importVmAsVappTemplate
Content-type: application/vnd.vmware.admin.importVmAsVAppTemplateParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ImportVmAsVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   name="ImportedWin2K8"
   sourceMove="false">
   <VmMoRef>vm-43</VmMoRef>
   <Vdc
      href="http://vcloud.example.com/api/vdc/2" />
   <Catalog
      href="http://vcloud.example.com/api/catalog/32" />
</ImportVmAsVAppTemplateParams>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.vcloud.vAppTemplate+xml
...
<VAppTemplate ...
   status="0"
   name="ImportedWin2K8"
   type="application/vnd.vmware.vcloud.vAppTemplate+xml"
   href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-64" ... >
   ...
   <Description />
   <Tasks>
      <Task
         operation=”Busy Virtual Application Template Win2K8 ”>
         ...
      </Task>
   </Tasks>
</VAppTemplate>
```

Configure the Password of an Imported Virtual Machine

When you are importing virtual machines from vCenter Server, or when you are synchronizing catalogs that have external subscriptions, the password properties values of the imported virtual machines are removed for security reasons. To configure a password of an imported virtual machine, you can update the `producSection` element of the virtual machine.

Synchronizing a vApp template from an externally published catalog imports an identical copy of the vApp template to the subscriber side, but the password properties values, if any, are removed. Importing a virtual machine from vCenter Server also removes any password properties values. After the import you can use the VMware Cloud Director API to configure the password of the virtual machine.

You configure the password once. Further catalog synchronisations keep the configured password.

Procedure

1 Retrieve an XML representation of the virtual machine.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/vAppTemplate/vm-33
   ```

2 Retrieve a list of the `ProductSection` elements of the virtual machine.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/vAppTemplate/vm-33/producSections
   ```

   The response contains a `ProductSectionList` element.

3 Copy and modify the retrieved `ProductSectionList` by adding the password value in the `Property` element.

4 PUT the `ProductSectionList` element that you modified in Step 3 to the `producSection` element of the virtual machine.

   ```
   PUT https://vcloud.example.com/api/vAppTemplate/vm-33/producSections
   ```

Example: Configure the Password of an Imported Virtual Machine

Request:

```
PUT https://vcloud.example.com/api/vAppTemplate/vm-32/producSections
Content-Type: application/*+xml;version=31.0
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProductSectionList
...
   <ovf:ProductSection ovf:class="" ovf:instance="" ovf:required="true">
        <ovf:Info>Information about the installed software</ovf:Info>
        <ovf:Property
         ovf:key="Password_Label"
         ovf:password="true"
         ovf:qualifiers="MinLen(1)"
         ovf:type="string"
         ovf:userConfigurable="true"
         ovf:value="My_Password">
            <ovf:Label>Password Label</ovf:Label>
        </ovf:Property>
   </ovf:ProductSection>
</ProductSectionList>
```

Response:

```
202 Status Code (The task that makes the change to the product section is passed back)
Content-Type: application/*+xml;version=31.0
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Task
   xmlns="http://www.vmware.com/vcloud/v1.5"
   ...
   operationName="templateUpdateVm"
   ...
   type="application/vnd.vmware.vcloud.task+xml">
   ...
</Task>
```

Relocate a Virtual Machine to a Different Datastore

If the datastore that contains a virtual machine has been disabled by the system administrator or is no longer associated with virtual machine's designated storage profile, you must update the `Vm` element that represents the virtual machine. That update revalidates the storage profile and relocates the virtual machine if necessary.

Every Vm element includes a StorageProfile element. The value of the href attribute of that element is a reference to the virtual machine's storage profile. The initial value of this attribute is inherited from the VDC that contains it unless you specify the value when the virtual machine is created. To change the value, you must update the entire Vm element that contains it.

Note When the system administrator changes the datastore that stores a virtual machine, you must update the Vm element as shown in Update the Storage Profile for a Virtual Machine, but leave the href of the current StorageProfile unchanged. This action, which replaces the deprecated relocate action, forces revalidation of the existing storage profile. If the current datastore is disabled or no longer supports the specified storage profile, the system relocates the virtual machine to a different datastore that supports the referenced storage profile. After the returned Task completes, the validation and, if necessary, relocation is complete.

Prerequisites

Verify that you are logged in to the VMware Cloud Director API as an administrator or the object owner.

Procedure

1 Retrieve the Vm element.

  Make a GET request to the URL in the value of the href attribute of the Vm.

2 Modify the retrieved Vm to change the StorageProfile reference.

  Request bodies must contain all required elements and attributes, even if you are not changing their values. Because optional elements and attributes typically revert to default values if they are omitted or empty, it is a best practice to include optional elements in request bodies that modify existing objects. Link elements and href attributes from responses do not need to be included in modified sections. Some elements and attributes are read-only and cannot be modified. See the schema reference for details.

3 Update the Vm with your modifications.

  a Find the Link element in the Vm where rel="edit".

  b Make a PUT request to the URL in that link's href attribute value, and supply the modified Vm as the request body.

  The response to this request is a Task element that tracks the relocation of the virtual machine to a datastore in the new storage profile. When the task is complete, the virtual machine has been relocated.

Example: Update the Storage Profile for a Virtual Machine

This example shows a Vm element containing a StorageProfile. The actual update operation requires the entire Vm element, including the StorageProfile, in the request body. Only a small part of the element appears in this example.

Request:

```
PUT https://vcloud.example.com/api/vApp/vm-4
Content-type: application/vnd.vmware.vcloud.vm+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Vm ...>
...
 <StorageProfile
      type="application/vnd.vmware.vcloud.vdcStorageProfile+xml"
      name="Gold"
      href="https://vcloud.example.com/api/vdcStorageProfile/3" />

</Vm>
```

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... operation="Updating Virtual Application Linux FTP server (7)" ...>
...
</Task>
```

Migrate Tenant Storage

You can migrate all vApps, independent disks, and catalog items of one or more organizations from one or more datastores to different datastores.

Before you decommission a datastore, you must migrate all the items stored on that datastore to a new datastore. You might also want to migrate an organization to a new datastore that has more storage capacity or uses a newer storage technology such as VMware vSAN.

Important Tenant storage migration is a resource-intensive operation that can run for a long time, especially when there are many assets to migrate. For more information about migrating tenant storage, see https://kb.vmware.com/kb/2151086.

Prerequisites

- For each storage policy containing a source datastore that you want to migrate, verify that there is at least one destination datastore to which to migrate. You can create destination datastores or use existing ones. For further information about determining the datastores in the storage policies used by the target organizations, see the *vSphere Storage* documentation.
- Create a VMware Cloud Director API session as a **system administrator** or with a role that has the **Organization: Migrate Tenant Storage** right.

Procedure

1 Retrieve the XML representation of the vSphere platform extensions.

Use a request like this one.

```
GET https://vcloud.example.com/api/admin/extension
```

2 Examine the response to locate the `Link` element that contains the URL for migrating tenant storage.

```
<Link
   type="application/vnd.vmware.cloud.tenantMigrationParams+xml"/>"
   rel="add"
   href="https://vcloud.example.com/api/admin/extension/action/migrateTenant"/>
```

3 Create a `TenantMigrationParams` element that includes the source and the target datastores and references one or more organizations to migrate.

4 POST the `TenantMigrationParams` element you created in Step 9 to the URL described in Step 8.

See the request portion of Migrate Tenant Storage.

Example: Migrate Tenant Storage

This example migrates items owned by organization with reference `https://vcloud.example.com/api/org/26` from datastore `https://vcloud.example.com/api/admin/extension/datastore/100` to datastore `https://vcloud.example.com/api/admin/extension/datastore/200`.

Request:

```
POST https://vcloud.example.com/api/admin/extension/action/migrateTenant
Content-type: application/vnd.vmware.cloud.tenantMigrationParams+xml
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vmext:TenantMigrationParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   name="example">
   <Description>Example migration params</Description>
   <vmext:orgs>
      <Org href="https://vcloud.example.com/api/org/26" />
   </vmext:orgs>
   <vmext:sourceDatastores>
      <vmext:Datastore
         href="https://vcloud.example.com/api/admin/extension/datastore/100" />
   </vmext:sourceDatastores>
   <vmext:targetDatastores>
```

```
      <vmext:Datastore
         href="https://vcloud.example.com/api/admin/extension/datastore/200" />
   </vmext:targetDatastores>
</vmext:TenantMigrationParams>
```

The response includes a `Task` that tracks the migration.

Response:

```
202 Accepted
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task ... >
   ...
</Task>
```

Retrieve the vSphere URL of an Object

If you know the `VimObjectType` and `MoRef` of an object represented in the VMware Cloud Director API, you can use that information to retrieve a URL that you can use to access the object with the vSphere Web Client.

Using the vSphere Web Client to examine an object VMware Cloud Director uses can help a system administrator diagnose problems with resource consumption and allocation. To retrieve the vSphere URL of an object, you must construct a request URL in the following format.

```
API-URL/admin/extension/vimServer/id/vimObjType/vimObjMoref/vSphereWebClientUrl
```

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122. The `vimServer` object that has this *id* must be the one that hosts the object that *vimObjType* and *vimObjMoref* identify.
- *vimObjType* is the vSphere object type, expressed as one of the following strings:
  - CLUSTER_COMPUTE_RESOURCE
  - DATASTORE
  - DATASTORE_CLUSTER
  - DV_PORTGROUP
  - DV_SWITCH
  - FOLDER
  - HOST
  - NETWORK
  - RESOURCE_POOL
  - STORAGE_PROFILE (available from vCenter 5.1 and later)

  - VIRTUAL_MACHINE
- *vimObjMoref* is the vSphere managed object reference of the object, as returned by the VMware Cloud Director API.

For an example request URL, see the request portion of Retrieve the vSphere URL of a Resource Pool. All of the information you need to construct the URL is available when you retrieve the XML representation of any of the supported object types. See Retrieve a Resource Pool Set, Retrieve a List of Available Portgroups and Switches from a vCenter Server, and Retrieve a List of Storage Profiles from a vCenter Server.

Note See Mapping a VMware Cloud Director Object to a vSphere Object for a list of VMware Cloud Director objects and corresponding vSphere objects.

Prerequisites

This operation is restricted to system administrators.

Verify that the vSphere Web Client URL is enabled for all vCenter Server instances from which you want to retrieve the vSphere URL of an object. You can manage this feature on the **General** tab of the vCenter Server details page in the VMware Cloud Director Service Provider Admin Portal.

Procedure

1. Retrieve the `VimObjectType` and `MoRef` elements of the object, and the `VimServerRef` element of the vSphere server that hosts the object.
2. Construct a request URL that includes the `VimServerRef`, `VimObjectType`, and `MoRef` elements.

   You can use the value of the `href` attribute of the `VimServerRef` element as the `vimServer`/*id* part of the request URL.
3. Make a GET request to the URL you constructed in Step 2.

   The response contains a URL you can use with the vSphere Web Client. See Retrieve the vSphere URL of a Resource Pool.

Example: Retrieve the vSphere URL of a Resource Pool

This request retrieves the vSphere URL of one of the resource pools referenced in Retrieve a Resource Pool Set.

Request:

```
GET https://vcloud.example.com/api/admin/extension/vimServer/9/RESOURCE_POOL/resgroup-195/
vSphereWebClientUrl
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<vmext:VSphereWebClientUrl
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   ...>
   <vmext:URL>https://10.147.22.134:8443/vsphere-client/
#extensionId=vsphere.core.sp.summary;...
   </vmext:URL>
</vmext:VSphereWebClientUrl>
```

Use the URL that the `vmext:URL` element contains to access the object with the vSphere Web Client. The URL is truncated in this example.

Mapping a VMware Cloud Director Object to a vSphere Object

When using the vSphere Web Client to examine a VMware Cloud Director object, you can use this table to map the object to its representation in vSphere.

Table 8-2. Mapping a VMware Cloud Director Object to a vSphere Object

VMware Cloud Director Object	vSphere Object
Provider VDC:Hosts	Host
Provider VDC:Datastores	Datastore
Provider VDC:Storage Profiles	Storage Profile
Provider VDC:External Network (portgroup-backed)	Standard virtual switch
Provider VDC:External Network(VLAN-backed)	Distributed virtual switch
Provider VDC:Resource Pool (root resource pool or Cluster)	Cluster
Provider VDC:Resource Pool (sub-resource pool only)	Resource Pool
External Network (Portgroup-backed)	Standard virtual switch
Direct Org VDC Networks (Portgroup)	Standard virtual switch
External Network (VLAN-backed)	Distributed virtual switch
Direct Org VDC Networks (VLAN)	Distributed virtual switch
Network Pool (VLAN-backed)	Distributed virtual switch
Network Pools (Portgroup-backed)	Standard virtual switch
vCenter	vCenter Server
Resource Pool (root resource pool or Cluster)	Cluster
Resource Pool (sub-resource pool only)	Resource Pool
Host	Host

Table 8-2. Mapping a VMware Cloud Director Object to a vSphere Object (continued)

VMware Cloud Director Object	vSphere Object
Datastores	Datastore
Datastore Clusters	Datastore Cluster
Storage Profiles	N/A
vDSwitch	Distributed virtual switch
Port Group	Standard virtual switch
Home:vApp	Folder
vApp	Folder
vApp:vApp Diagram	Folder
vApp:vApp Diagram:VM	VM
vApp:VM List:VM	VM
vApp:Networking:vApp Networks	Distributed Port Group
VM	VM
Expired Items:vApps	Folder
Expired Items:vApp Templates	Folder
Catalog:vApp Template	Folder
Catalog:vApp Template:VM	VM
Catalog:vApp Template:Shadow VMs	VM
Organization VDC:vApp	Folder
Organization VDC:vApp Template	Folder
Organization VDC:Storage Profiles	Storage Profile
Organization VDC:Org VDC Networks (VLAN-backed)	Distributed virtual switch
Organization VDC:Org VDC Networks (Portgroup-backed)	Standard virtual switch
Organization VDC:Resource Pools (root resource pool or Cluster)	Cluster
Organization VDC:Resource Pools (sub-resource pool only)	Resource Pool

Configuring and Managing Multisite Deployments

To manage and monitor multiple, geographically distributed VMware Cloud Director installations or server groups and their organizations as single entities, you can use the VMware Cloud Director multisite feature.

Effective Implementation of a Multisite

When you associate two VMware Cloud Director sites, you enable the administration of the sites as a single entity. You also enable organizations at those sites to form associations with each other. See #unique_212. When an organization is a member of an association, organization users can use the VMware Cloud Director Tenant Portal to access organization assets at any member site, although each member organization and its assets are local to the site it occupies.

Note The sites must be with the same VMware Cloud Director API version, or one major version apart. For example, you can associate a VMware Cloud Director 10.1 (API version 34.0) site with a VMware Cloud Director site version 10.0, 10.1, 10.2 or 10.2.2, respectively API versions 33.0, 34.0, 35.0, or 35.2.

After you associate two sites, you can use the VMware Cloud Director API or the VMware Cloud Director Tenant Portal to associate organizations that occupy those sites. See the *VMware Cloud Director API Programming Guide* or the Configure and Manage Multisite Deployments topic in the *VMware Cloud Director Tenant Guide*.

A site or organization can form an unlimited number of associations with a peer, but each association includes exactly two members. Each site or organization must have its own private key. Association members establish a trust relationship by exchanging public keys, which are used to verify signed requests from one member to another.

Each site in an association is defined by the scope of a VMware Cloud Director server group (a group of servers that share a VMware Cloud Director database). Each organization in an association occupies a single site. The organization administrator controls access by organization users and groups to assets at each member site.

Site Objects and Site Associations

The installation or upgrade process creates a `Site` object that represents the local VMware Cloud Director server group. A system administrator whose authority extends to more than one VMware Cloud Director server group can configure those server groups as an association of VMware Cloud Director sites.

Site Names

Each `Site` object is created with a name attribute that is a UUID.

```
GET https://Site-B.example.com/api/site
...
<Site name="b5920690-fe13-4c31-8e23-9e86005e7a7b" ...>
   ...
   <RestEndpoint>https://Site-A.example.com/api/org/Org-A</RestEndpoint>
   <RestEndpointCertificate>-----BEGIN CERTIFICATE-----
      MIIDDTCCAfWgAwIBAgI...Ix0eSE= -----END CERTIFICATE-----
   </RestEndpointCertificate>
   ...
</Site>
```

While there is no requirement that the site name matches the hostname in the API endpoint, a system administrator can update the site name as an administrative convenience for VMware Cloud Director API users, with a request like this one:

```
PUT https://Site-B.example.com/api/site
content-type: application/vnd.vmware.vcloud.site+xml
...
<Site name="Site-B" ...>
   ...
   <RestEndpoint>https://Site-A.example.com/api/org/Org-A</RestEndpoint>
   <RestEndpointCertificate>-----BEGIN CERTIFICATE-----
      MIIDDTCCAfWgAwIBAgI...Ix0eSE= -----END CERTIFICATE-----
   </RestEndpointCertificate>
   ...
</Site>
```

The `Site` element in the request body must retain the formatting in which it was returned by the `GET .../site` request. Additional characters, particularly carriage-returns, line feeds, or spaces, before or after the certificates can cause the system to return a bad request exception.

Associations of Organizations

After site association is complete, **organization administrators** at any member site can begin associating their organizations.

Note You cannot associate a `System` organization with a tenant organization. The `System` organization at any site can be associated only with the `System` organization at another site.

Authorization Headers and Request Fanout

The `Session` response to a successful login request includes an `X-VMWARE-VCLOUD-ACCESS-TOKEN` header whose value is an encoded key that you can use, and the value of the `X-VMWARE-VCLOUD-TOKEN-TYPE` header, to construct a JWT `Authorization` header to include in subsequent requests in place of the deprecated `x-vcloud-authorization` header, which does not authenticate you to association members. See Create a VMware Cloud Director API Session for more information about logging in to the VMware Cloud Director API.

You can make requests that fan out to multiple association members by specifying the `multisite=`*value* pair in the `Accept` header. If you want the request to fan out, the *value* can be `global` or a colon-separated list of location IDs. For information about obtaining the location IDs, see Authorized Locations. When you set the *value* to `local`, the request does not fan out but includes multisite metadata included on fan-out.

For example, when you make a request such as a query that retrieves lists of resources from an association of organizations, you can specify the `multisite=global` pair in the `Accept` header. By specifying the `multisite=global` pair, you fan out the request to all association members and return an aggregated list.

```
Accept: application/*;version=30.0;multisite=global
```

You can specify a colon-separated list of location IDs, for example, `multisite=`*ID-a:ID-b:ID-x*. Unless you include this value in the `Accept` header, the request returns only those resources owned by the organization that is the target of the request. Unless you are making a request to the same organization that you authenticated to, you must also include a `X-VMWARE-VCLOUD-AUTH-CONTEXT` header that specifies the name of the organization that will fulfill your request.

When an authentication request includes the `multisite=`*value* pair, the response includes `Link` elements if the request failed at any association member. The category of the failure is represented by the `rel` value of the link.

rel="fanout:failed"

Member status was `ACTIVE` but authentication at the member failed for some other reason.

rel="fanout:skipped"

Authentication at the member was skipped because the association status was `ASYMMETRIC` or `UNREACHABLE`.

The failed or skipped request URL is in the `href` the value of the `Link`.

Authorized Locations

When you authenticate to a site that is a member of an association, the `Session` response includes an `AuthorizedLocations` element that provides VMware Cloud Director API and VMware Cloud Director Tenant Portal endpoints for other association members. In this example:

- `Site-A.example.com` and `Site-B.example.com` are associated.

- The user logs in to Site-A as a **system administrator**.

Request:

```
POST https://Site-A.example.com/api/sessions
Authorization: Basic ...
Accept: application/*;version=30.0
...
```

Response:

```
200 OK
...
X-VMWARE-VCLOUD-ACCESS-TOKEN: eyJhbGciOiJSUzI1NiJ9....Rn4Xw
X-VMWARE-VCLOUD-TOKEN-TYPE: Bearer
Content-Type: application/vnd.vmware.vcloud.session+xml;version=30.0;multisite=global
...
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Session ...
   ...
   <AuthorizedLocations>
      <Location>
         <LocationId>a93c9db9-7471-3192-8d09-a8f7eeda85f9@9a41...
         </LocationId>
         <SiteName>Site-A</SiteName>
         <OrgName>System</OrgName>
         <RestApiEndpoint>https://site-a.example.com
         </RestApiEndpoint>
         <UIEndpoint>https://site-a.example.com
         </UIEndpoint>
         <AuthContext>System</AuthContext>
      </Location>
      <Location>
         <LocationId>a93c9db9-7471-3192-8d09-a8f7eeda85f9@4f56...
         </LocationId>
         <SiteName>Site-B</SiteName>
         <OrgName>System</OrgName>
         <RestApiEndpoint>https://site-b.example.com
         </RestApiEndpoint>
         <UIEndpoint>https://site-b.example.com
         </UIEndpoint>
         <AuthContext>System</AuthContext>
      </Location>
   </AuthorizedLocations>
</Session>
```

User and Group Identities

Associations of sites and organizations must agree to use the same identity provider (IDP). User and group identities for all organizations in the association must be managed through this IDP.

Associations are free to choose the IDP that works best for them. See #unique_213.

Starting with VMware Cloud Director 10.4.1, service accounts can manage and monitor multiple, geographically distributed VMware Cloud Director installations or server groups and their organizations as single entities by using the multisite feature. If a service account is making a request to a different organization from the one that it is authenticated to, verify that the service account exists on the associated organization and that it has the same name and software ID. See #unique_214.

Site Access Control for Organization Users and Groups

Organization administrators can configure their IDP to generate user or group access tokens that are valid at all member sites, or valid at only a subset of member sites. While user and group identities must be the same in all member organizations, user and group rights are constrained by the roles those users and groups are assigned in each member organization. Assignment of a role to a user or group is local to a member organization, as are any custom roles you create.

Load Balancer Requirements

Effective implementation of a multisite deployment requires you to configure a load balancer that distributes requests arriving at an institutional endpoint such as `https://vcloud.example.com` to the endpoints for each member of the site association (for example, `https://us.vcloud.example.com` and `https://uk.vcloud.example.com`). If a site has more than one cell, you must also configure a load balancer that distributes incoming requests across all its cells, so that a request to `https://us.vcloud.example.com` can be handled by `https://cell1.us.vcloud.example.com`, `https://cell2.us.vcloud.example.com`, and so on.

Note You must use the global load balancer, in this case `https://vcloud.example.com`, only for UI access. If you develop your own scripts or programs that use the REST API, those calls must target a particular site.

Starting with VMware Cloud Director 10.3, all client requests that arrive at the load-balancing endpoint for a multisite deployment are redirected. When a request arrives at the load-balancing endpoint, even if the site where the request arrives is the correct one, a redirect is issued and reflected in the user-visible URL to specify that the request was directed to the correct location.

For example, you can have a deployment consisting of two sites - `https://site1.vcloud.example.com` and `https://site2.vcloud.example.com` - behind a global load-balancing endpoint `https://us.vcloud.example.com`. Starting with VMware Cloud Director 10.3, when a request arrives at the load-balancing endpoint for an organization that is located at site 1 - `https://us.vcloud.example.com/org1`, if the request lands at site 1, then site issues a redirect to itself by forwarding the request to `https://site1.vcloud.example.com/org1`. VMware Cloud Director 10.2.x and earlier versions do not issue redirects when a load balancer receives a request for an organization that is located at the same place and the request is serviced through the public endpoint's URL `https://us.vcloud.example.com/org1`.

Network Connectivity Requirements

If you want to use the multisite feature, each cell at each site must be able to make REST API requests to the REST API endpoints of all sites. If you use the examples from the Load Balancer Requirements section, `cell1.us.vcloud.example.com` and `cell2.us.vcloud.example.com` must be able to reach the REST API endpoint for `uk.example.com`. The reverse is true for all cells under `uk.example.com`. This means that a cell must also be able to make REST API calls to it's own REST API endpoint, so `cell1.us.vcloud.example.com` must be able to make a REST API call to `https://us.vcloud.example.com`.

Making REST API requests to the REST API endpoints of all sites is necessary for of REST API fanout. For example, if the UI or an API client makes a multisite request to get a page of organizations from all sites and `cell1.us.vcloud.example.com` handle the request. The cell `cell1` must make a REST API call to get a page of organizations from each site using the REST API endpoint configured for that site. When all sites return their page of organizations, `cell1` collates the results and returns a single page of results containing the data from all other sites.

Sites and Certificates

When a site is associated with other sites, if you update its certificate, you might have to let the other sites know of the change. If you do not let the other sites know about the certificate change, the multisite fanout might be impacted.

If you are replacing a certificate on a site with a valid, well-signed certificate, then you do not need to inform the other sites. Because the certificate is valid and well-signed, the cells at the other sites can continue connecting to it in a secure manner without interruption.

If you are replacing a certificate on a site with a self-signed certificate, or if there is some other problem with the certificate that prevents automatic trust, then other sites need to know. For example, if the certificate expires, you must let the other sites know. At each of the other sites, you must upload the certificate into the **Trusted Certificates** in the Service Provider Admin Portal. See #unique_215. When you import the certificate, the site where the certificate is uploaded can trust the site getting the new certificate.

Note You can import these certificates to the Trusted Certificates of the other sites before you install them at the remote site. This ensures no interruptions in communication because both the old certificate and the new certificate are in the Trusted Certificates pool. You do not have to reassociate the sites.

Association Member Status

After you create an association of sites or organizations, the local system periodically retrieves the status of each remote association member and updates that status in the local site's VMware Cloud Director database.The member status is visible in the `Status` element of an `SiteAssociationMember` or `OrgAssociationMember`. The `Status` element can have one of three values:

`ACTIVE`

> The association has been established by both parties, and communication with the remote party was successful.

`ASYMMETRIC`

> The association has been established at the local site, but the remote site has not yet reciprocated.

`UNREACHABLE`

> An association has been created by both parties, but the remote site is not currently reachable on the network.

In the Service Provider Admin Portal and Tenant Portal the statuses appear as `Connected`, `Partially Connected`, and `Unreachable`.

The member status "heartbeat" process runs with the identity of the multisite system user, a local VMware Cloud Director user account created in the System organization during VMware Cloud Director installation. Although this account is a member of the System organization, it does not have system administrator rights. It has only a single right, `Multisite: System Operations`, which gives it permission to make a VMware Cloud Director API request that retrieves the status of the remote member of a site association.

Associate Two Sites

A system administrator whose authority extends to more than one VMware Cloud Director server group can configure those server groups as an association of VMware Cloud Director sites.

To create an association between two sites (we'll call them Site-A and Site-B here), you must be a system administrator at both sites so that you can log in to each site to retrieve the information required to associate the site with another site, then make a POST request to each site supplying the other site's association information.

Important The process of associating two sites can be logically decomposed into two complementary pairing operations. The first operation (in this example) pairs Site-A with Site-B. You must then go on to pair Site-B with Site-A. Until both pairings are complete, the heartbeat process reports that the association is incomplete (showing a `Status` value of `ASYMMETRIC`).

Prerequisites

- This operation is restricted to system administrators.
- You must be a system administrator at both sites.

Procedure

1 Log in to Site-A and retrieve its site association data.

Site association data is contained in the `SiteAssociations` element of the `Site` object.

```
GET https://Site-A.example.com/api/site
...
<Site
   name="Site-A"
   id="urn:vcloud:site:94b78377-a3f8-4970-8e44-662d209f35f7"
   href="https://Site-A.example.com/api/site">
   ...
   <Link
      rel="down"
      href="https://Site-A.example.com/api/site/associations" ... />
   ...
</Site>
```

The `SiteAssociations` element includes a link of the form:

```
<Link
   rel="down"
   href="https://Site-A.example.com/api/site/associations/localAssociationData"/>
```

Make a GET request to this link to retrieve the `SiteAssociationMember` element that contains the information required when associating this site with another site.

```
GET https://Site-A.example.com/api/site/associations/localAssociationData
...
<SiteAssociationMember ...>
   <Link ...>
   ...
   <RestEndpoint>https://Site-A.example.com</RestEndpoint>
   <RestEndpointCertificate>-----BEGIN CERTIFICATE-----
      MIIDDTCCAfWgAwIBAgI...Ix0eSE= -----END CERTIFICATE-----
   </RestEndpointCertificate>
   <SiteId>urn:vcloud:site:94b78377-a3f8-4970-8e44-662d209f35f7</SiteId>
   <SiteName>Site-A</SiteName>
   <PublicKey>-----BEGIN PUBLIC KEY-----
      MIIBIjANBgkqhki...QQIDAQAB -----END PUBLIC KEY-----
   </PublicKey>
</SiteAssociationMember>
```

Save this response. You'll need it in Step 3.

2 Log in to Site-B and retrieve its `add` link for associations.

This link is contained in the `SiteAssociations` element of the `Site` object .

```
GET https://Site-B.example.com/api/site
...
<Site
   name="Site-B"
   ...
   <Link
      rel="down"
      href="https://Site-B.example.com/api/site/associations" ... />
   ...
</Site>
```

Retrieve the `SiteAssociations` to see this link.

```
GET https://Site-B.example.com/api/site/associations
...
<SiteAssociations
   ...
   <Link
      rel="add"
      href="https://Site-B.example.com/api/site/associations" ... />
   ...
</Site>
```

3 POST the `SiteAssociationMember` response body you retrieved in Step 1 to the `siteAssociations` link for Site-B.

You can include the entire response body. Any `Link` elements from the response body are ignored in a request. For brevity, this example omits the `Link` elements.

```
POST https://Site-B.example.com/api/site/associations
content-type: application/vnd.vmware.admin.siteAssociation+xml
...
<SiteAssociationMember ...>
   <RestEndpoint>https://Site-A.example.com</RestEndpoint>
   <RestEndpointCertificate>-----BEGIN CERTIFICATE-----
      MIIDDTCCAfWgAwIBAgI...Ix0eSE= -----END CERTIFICATE-----
   </RestEndpointCertificate>
   <SiteId>urn:vcloud:site:94b78377-a3f8-4970-8e44-662d209f35f7</SiteId>
   <SiteName>Site-A</SiteName>
   <PublicKey>-----BEGIN PUBLIC KEY-----
      MIIBIjANBgkqhki...QQIDAQAB -----END PUBLIC KEY-----
   </PublicKey>
</SiteAssociationMember>
```

The response is a `Task`.

When the task succeeds, Site-A has been paired with Site-B. Until you pair Site-B with Site-A, the `Status` of the association is incomplete (`ASYMMETRIC`).

```
<SiteAssociationMember ...>
   ...
   <Status>ASYMMETRIC</Status>
</SiteAssociationMember>
```

What to do next

Repeat this procedure, modified as needed to retrieve the `SiteAssociationMember` content from Site-B and POST it to Site-A. The multisite heartbeat process confirms the network connection between the sites and then sets the `Status` value of each `SiteAssociationMember` to `ACTIVE`.

Important If you plan to create stretched networks across the associated sites, ensure that there are no MAC address conflicts. If the two sites are configured with the same installation ID, you must regenerate the MAC addresses in one of the sites. For information about regenerating MAC addresses, see the cell management tool reference in the *VMware Cloud Director Administrator's Guide*.

Associate Organizations at Member Sites

After site association is complete, organization administrators at any member site can begin associating their organizations.

To create an association between two organizations (we'll call them Org-A and Org-B here), you must be a system administrator at both sites so that you can log in to each site to retrieve the information required to associate the site with another site, then make a POST request to each site supplying the other site's association information.

You cannot associate a `System` organization with a tenant organization. The `System` organization at any site can be associated only with the `System` organization at another site.

Important The process of associating two organizations can be logically decomposed into two complementary pairing operations. The first operation (in this example) pairs Org-A at Site-A with Org-B at Site-B. You must then go on to pair Org-B at Site-B with Org-A at Site-A. Until both pairings are complete, the association is incomplete. Until both pairings are complete, the heartbeat process reports that the association is incomplete (showing a `Status` value of `ASYMMETRIC`).

Prerequisites

- The sites occupied by the organizations must be associated. See Associate Two Sites.
- You must be a system administrator at both sites or an organization administrator of both organizations.

Procedure

1 Log in to Site-A and retrieve the association data from Org-A.

Site association data is contained in the `OrgAssociations` element of the `AdminOrg` object.

```
GET https://Site-A.example.com/admin/org/30
...
<AdminOrg name="Org-A" ...>
   ...
   <Link
      rel="down"
      href="https://Site-A.example.com/api/admin/org/30/associations" ... />
   ...
</AdminOrg>
```

The `OrgAssociations` element includes a link of the form:

```
<Link
   rel="down"
   href="https://Site-A.example.com/api/admin/org/30/associations/localAssociationData"/>
```

Make a GET request to this link to retrieve the `OrgAssociationMember` element that contains the information required when associating this organization with another with an organization at an associated site.

```
GET https://Site-A.example.com/api/admin/org/30/associations/localAssociationData
...
<OrgAssociationMember ...>
   <Link ...>
   ...
   <RestEndpointCertificate>-----BEGIN CERTIFICATE-----
      MIIDDTCCAfWgAwIBAgI...Ix0eSE= -----END CERTIFICATE-----
   </RestEndpointCertificate>
   <SiteId>urn:vcloud:org:94b78377-a3f8-4970-8e44-662d209f35f7</SiteId>
   <OrgId>urn:vcloud:org:f47c905d-6e7f-4ef2-a7ec-7136660bdc4f</OrgId>
   <OrgName>Org-A</OrgName>
   <OrgPublicKey>-----BEGIN PUBLIC KEY-----
      MIIBIjANBgkqhki...QQIDAQAB -----END PUBLIC KEY-----
   </OrgPublicKey>
</OrgAssociationMember>
```

Save this response. You'll need it in Step 3.

2 Log in to Org-B and retrieve its `add` link for associations.

This link is contained in the `OrgAssociations` element of the `AdminOrg` object .

```
GET https://Site-B.example.com/admin/org/40
...
<AdminOrg name="Org-B" ...>
   ...
   <Link
```

```
      rel="down"
      href="https://Site-B.example.com/api/admin/org/40/associations" ... />
   ...
</AdminOrg>
```

Retrieve the `OrgAssociations` to see this link.

```
GET https://Site-B.example.com/admin/org/40/associations
...
<OrgAssociations
   ...
   <Link
      rel="add"
      href="https://Site-B.example.com/api/admin/org/40/associations" ... />
   ...
</OrgAssociations>
```

3 POST the `OrgAssociationMember` response body you retrieved in Step 1 to the `orgAssociations` link for Site-B.

You can include the entire response body. Any `Link` elements from the response body are ignored in a request. For brevity, this example omits the `Link` elements.

```
POST https://Site-B.example.com/api/admin/org/40/associations
content-type: application/vnd.vmware.admin.organizationAssociation+xml
...
<OrgAssociationMember ...>
   ...
   <RestEndpointCertificate>-----BEGIN CERTIFICATE-----
      MIIDDTCCAfWgAwIBAgI...Ix0eSE= -----END CERTIFICATE-----
   </RestEndpointCertificate>
   <SiteId>urn:vcloud:org:94b78377-a3f8-4970-8e44-662d209f35f7</SiteId>
   <OrgId>urn:vcloud:org:f47c905d-6e7f-4ef2-a7ec-7136660bdc4f</OrgId>
   <OrgName>Org-A</OrgName>
   <OrgPublicKey>-----BEGIN PUBLIC KEY-----
      MIIBIjANBgkqhki...QQIDAQAB -----END PUBLIC KEY-----
   </OrgPublicKey>
</OrgAssociationMember>
```

The response is a `Task`.

When the task succeeds, Org-A has been paired with Org-B.

What to do next

Repeat this procedure, modified as needed to retrieve the `OrgAssociationMember` content from Org-B and POST it to Org-A. This completes the association. The multisite heartbeat process confirms the network connection between the organizations and then sets the `Status` value of each `OrgAssociationMember` to `ACTIVE`.

Managing Dedicated vCenter Server Instances

With dedicated vCenter Server instances, you can use VMware Cloud Director as a central point of management (CPOM) for your vSphere environments.

When you add a vCenter Server instance to VMware Cloud Director, you can specify the purpose of the instance.

Dedicated vCenter Server

The infrastructure of an attached vCenter Server instance is encapsulated as a Software-Defined Data Center (SDDC) and is fully dedicated to a single tenant. You create a dedicated vCenter Server instance by activating the tenant access for that instance. After you activate the tenant access, you can publish a dedicated vCenter Server instance to a tenant.

Shared vCenter Server

The provider can use different resource pools of the vCenter Server instance across multiple provider VDCs and then allocate those resource pools to different tenants. A shared vCenter Server instance cannot be published to tenants.

None

The vCenter Server instance does not have any specific purpose.

VMware Cloud Director can act as an HTTP proxy server for the dedicated vCenter Server instances and the vCenter Server instances that do not have a set purpose.

With dedicated vCenter Server instances, you can use VMware Cloud Director as a central point of management for all your vSphere environments.

- You can dedicate the resources of a vCenter Server instance to a single tenant by publishing the corresponding dedicated vCenter Server only to its organization. The tenant does not share these resources with other tenants. The tenant can access this dedicated vCenter Server instance by using a UI or API proxy without a VPN required.
- You can use VMware Cloud Director as a lightweight directory to register all your vCenter Server instances.
- You can use VMware Cloud Director as an API endpoint for all your vCenter Server instances.

You can activate the tenant access and mark a vCenter Server instance as dedicated, during or after the attachment of the target vCenter Server instance to VMware Cloud Director. See #unique_184.

With an attached vCenter Server instance, you can create either a shared vCenter Server or a dedicated vCenter Server. If you created a shared vCenter Server instance, you cannot use this vCenter Server instance to create a dedicated vCenter Server, and the reverse.

You can create endpoints that tenants can use to access the underlying vSphere environment. The VMware Cloud Director credentials are for the proxied components that connect to vCenter Server. The vCenter Server instances have different credentials.

Dedicated vCenter Server instances in VMware Cloud Director remove the requirement for vCenter Server to be publicly accessible. To control the access, you can activate and deactivate the tenant access to an SDDC in VMware Cloud Director.

An endpoint is the access point to a component from an SDDC, for example, a vCenter Server instance, an ESXi host, or an NSX Manager instance. You can connect an endpoint to a proxy. By activating and deactivating a proxy, you can allow and stop the tenant access through that proxy.

Starting with VMware Cloud Director 10.2, if you use the API to query the dedicated vCenter Server and proxy entities and your tenant configuration supports multisite associations, VMware Cloud Director returns a multisite response. The results are from all available associations.

Creating and Managing Dedicated vCenter Server Instances

To create and manage dedicated vCenter Server instances and proxies, you can use the Service Provider Admin Portal or the VMware Cloud Director OpenAPI. For VMware Cloud Director OpenAPI, see *Getting Started with VMware Cloud Director OpenAPI* at https://developer.vmware.com/.

Important VMware Cloud Director requires a direct network connection to each dedicated vCenter Server instance. If the vCenter Server instance uses an external Platform Services Controller, VMware Cloud Director requires a direct network connection to the Platform Services Controller as well.

To use VMware OVF Tool in a proxied dedicated vCenter Server, VMware Cloud Director requires a direct connection to each ESXi host.

1 Create a dedicated vCenter Server instance.

   When you add a vCenter Server instance to the VMware Cloud Director environment, you can create a dedicated vCenter Server instance by activating the tenant access in the **Add vCenter Server** wizard. See #unique_219.

   Creating a dedicated vCenter Server instance also creates a default endpoint for it. While attaching the vCenter Server instance, you can also create a proxy. However, the default endpoint is not connected to any proxy by default. You must edit the default endpoint or create a new one to connect it to a proxy. See #unique_220.

   You can activate the tenant access of vCenter Server instances that are already added to VMware Cloud Director and do not have a specified use. See #unique_221. Activating the tenant access makes the vCenter Server instance available to be published to tenants.

2 Add a proxy.

You can create a proxy either when you attach a vCenter Server instance to VMware Cloud Director or later. If the vCenter Server instance uses an external Platform Services Controller, VMware Cloud Director creates a proxy for the Platform Services Controller as well. With parent and child proxies, you can hide certain proxies from the tenants or you can activate and deactivate groups of child proxies through their parent proxies. For information on creating a proxy after you add a vCenter Server instance to VMware Cloud Director, see #unique_222.

You can edit, activate, deactivate, and delete proxies from the **Proxies** tab under **vSphere Resources**.

Note When you add a proxy to a dedicated vCenter Server instance, you must upload the certificate and the thumbprint, so that tenants can retrieve the certificate and the thumbprint if the proxied component uses self-signed certificates.

To view and manage certificates and certificate revocation lists (CRLs), see #unique_223.

3 Get the certificate and the thumbprint of the created proxies, and verify that the certificate and the thumbprint are present and correct. See #unique_223.

4 Publish the dedicated vCenter Server instance to one or more organizations.

   You can publish a dedicated vCenter Server instance to a tenant and make it visible in the VMware Cloud Director Tenant Portal. In most cases, one vCenter Server instance should be published only to one tenant. See #unique_224.

5 To enable the tenants to access the dedicated vCenter Server instances and proxies from the VMware Cloud Director Tenant Portal, you must publish the **CPOM extension** plug-in to their organizations. See . See the *VMware Cloud Director Service Provider Admin Guide*.

Working with Object Metadata

9

The VMware Cloud Director API provides a general-purpose facility for associating user-defined metadata with an object. An administrator or object owner can use the `metadata` link in the object's representation to access an object's metadata.

Object metadata gives cloud operators and tenants a flexible way to associate user-defined properties (*`name=value`* pairs) with objects. Object metadata is preserved when objects are copied, and can be included in query filter expressions (see Add a Metadata Filter to a Query).

VMware Cloud Director API Object Metadata Links

The representation of any object that supports metadata includes a link that you can use to retrieve the object's `Metadata` element. This example shows the `metadata` link from a `VApp` element.

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.metadata+xml"
   href="https://vcloud.example.com/api/vApp/vapp-7/metadata"/>
```

The following elements can include a link to a `Metadata` element.

- `Catalog`
- `CatalogItem`
- `Disk`
- `Media`
- `OrgVdcNetwork`
- `ProviderVdc`
- `ProviderVdcStorageProfile`
- `VApp`
- `VAppTemplate`
- `Vdc`
- `VdcStorageProfile`

- Vm

A Metadata element can contain up to 1024 MetadataEntry elements (*name=value* pairs) that the object owner or an administrator can create, retrieve, update, and delete. It also contains a group of *name=value* pairs that are under the control of the system administrator.

VMware Cloud Director API Object Metadata Contents

Each MetadataEntry includes a single *name=value* pair, represented in its Key and TypedValue elements. Key element contents specify key names. A key name is a UTF-8 (Unicode) string encoded as described in RFC3986 (pct-encoded).

The key name must be unique within the scope of the containing Metadata element. Because key names are implicitly qualified by the Domain value of the containing MetadataEntry, the two key names in this example are considered to be unique.

```
<Metadata
 ...
   <MetadataEntry>
      <Domain
         visibility="READONLY">SYSTEM</Domain>
      <Key>Foo</Key>
      ...
   </MetadataEntry>
  <MetadataEntry>
      <Key>Foo</Key>
      ...
   </MetadataEntry>
</Metadata>
```

See Access Control for VMware Cloud Director API Object Metadata for more information about the Domain element.

The type of a Value is expressed in the xsi:type attribute of the containing TypedValue element. Values have various restrictions, based on their type.

Table 9-1. Metadata TypedValue Types

Type Name	Restrictions on Value	Size of Value
MetadataStringValue	UTF-8 character set. Strings longer than 1000 characters cannot be searched for in a query.	Depends on length of string.
MetadataNumberValue	Must be a signed 8-byte integer.	8 bytes

Table 9-1. Metadata TypedValue Types (continued)

Type Name	Restrictions on Value	Size of Value
MetadataDateTimeValue	UTC date and time in the format defined by http://www.w3.org/TR/xmlschema-2/#dateTime. For example, `2012-06-18T12:00:00-05:00`	8 bytes
MetadataBooleanValue	Must be one of: 1, 0, true, false	1 byte

The following rules apply when you update a `Metadata` element.

- When the content of a `Key` element in the update does not match the content of an existing `Key` element in the same `Domain`, the `MetadataEntry` containing that `Key` is added to the `Metadata` element.
- When the content of `Key` element in the update matches the content of an existing `Key` element in the same `Domain`, the `MetadataEntry` containing that `Key` is replaced.

Access Control for VMware Cloud Director API Object Metadata

The `Domain` element of a `MetadataEntry` controls access to that entry by users and system administrators. There are two access domains, only one of which can be specified explicitly.

GENERAL

This is the default access domain for metadata, and cannot be explicitly specified in a `Domain` element. A `MetadataEntry` that does not include a `Domain` element is considered to be in the `GENERAL` domain. User access to metadata in the `GENERAL` domain is controlled by the user's rights to the object to which the `Metadata` is attached. The owner of the object can create, read, update, or delete any `MetadataEntry` in the `GENERAL` domain.

SYSTEM

To be placed in the `SYSTEM` domain, a `MetadataEntry` must include a `Domain` element with a value of `SYSTEM`. Metadata in the `SYSTEM` domain can be created, updated, and deleted only by a system administrator. User access to metadata in the `SYSTEM` domain is controlled by the value of the `visibility` attribute of the `Domain` element.

Table 9-2. Domain Visibility Attribute Values

Value	Meaning
PRIVATE	The metadata is visible to system administrators only.
READONLY	the metadata is visible to system administrators and the object owner.

VCENTER

Object metadata in the VCENTER domain can be applied to Vm objects, including Vm objects contained in VAppTemplate objects, by users in roles that have the right vApp: Allow metadata mapping domain to vCenter. The system uses metadata in this domain to deploy virtual machines on specific ESXi hosts that your service provider configures. Your service provider establishes the set of legal values for metadata in this domain. See Specifying Advanced Virtual Machine Settings with ExtraConfig Elements.

Note Object metadata in the VCENTER domain is not displayed in the VMware Cloud Director Web Console.

VMware Cloud Director API Object Metadata Limits

The following limits apply to VMware Cloud Director API object metadata:

Metadata key size

The contents of a Key element in a MetadataEntry cannot exceed 256 UTF-8 characters.

Metadata size

In VMware Cloud Director 10.2, the size of all Metadata for an object, computed as the sum of all Key and TypedValue UTF-8 strings in all MetadataEntry elements in the GENERAL domain, cannot exceed 128 KB. An additional 16 KB of MetadataEntry content can be created in the SYSTEM domain.

Starting with VMware Cloud Director 10.2.1, the size of all Metadata for an object, computed as the sum of all Key and TypedValue UTF-8 strings in all MetadataEntry elements are configurable. The default size is 128 KB for all domains.

You can configure the size through the configuration properties in the format com .vmware.vcloud.metadata.impl.config.<domain>.<target>. You can change the configuration properties at runtime by using JMX configuration MBean.

```
com.vmware.vcloud.metadata.impl.config.GENERAL.maxBytesPerEntity = 262144
com.vmware.vcloud.metadata.impl.config.SYSTEM.maxBytesPerEntity = 512
```

MetadataEntry limit

In VMware Cloud Director 10.2, the total metadata associated with an object cannot exceed 1024 `MetadataEntry` elements in the `GENERAL` domain and 128 `MetadataEntry` elements in the `SYSTEM` domain.

Starting with VMware Cloud Director 10.2.1, you can configure the total `MetadataEntry` items associated with an object.

```
com.vmware.vcloud.metadata.impl.config.GENERAL.maxItemsPerEntity = 1000
com.vmware.vcloud.metadata.impl.config.SYSTEM.maxItemsPerEntity = 50
```

Read the following topics next:

- Retrieve or Update a Metadata Element
- Retrieve or Update a Metadata Value

Retrieve or Update a Metadata Element

An administrator or the owner of an object can create, retrieve, or update the object's `Metadata` element. This element contains all object metadata. Operations that modify it merge the modifications with existing contents.

When you create an object, its representation contains an empty `Metadata` element. An administrator or the object owner can add metadata by updating the `Metadata` element with new `MetadataEntry` elements. Each of these elements contains a `Key` and a `TypedValue`. The contents of the `Key` element define the key name, which must be unique within the scope of the object's metadata. You can modify the value associated with an existing key. See Retrieve or Update a Metadata Value.

Note The `Key` element cannot contain a semicolon character (;). In addition, several other character sequences are not allowed, or not allowed in certain positions.

Table 9-3. Content Restrictions for `Key`

Cannot Contain	Cannot Start with	Cannot End with
`/../`	`./`	`/.`
`/./`	`../`	`/..`

Prerequisites

- Verify that you are logged in to the VMware Cloud Director API as an administrator or the object owner.
- Retrieve the object's `Metadata` element. See Retrieve or Update a Metadata Element

Procedure

1 Retrieve the representation of the object.

Examine the response to find its metadata link. This example shows the `metadata` link from an `AdminOrg`.

```
<Link
   rel="down"
   type="application/vnd.vmware.vcloud.metadata+xml"
   href="https://vcloud.example.com/api/admin/vApp/vapp-7/metadata"/>
```

2 Retrieve the `Metadata` element.

If the object has no metadata, the element contains only a `rel="add"` link that you can use to add metadata and a `rel="up"` link that references the containing object, as shown in this example.

```
<Metadata
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.vcloud.metadata+xml"
   href="https://vcloud.example.com/api/vApp/vapp-7/metadata"
   ... >
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vApp+xml"
      href="https://vcloud.example.com/api/vApp/vapp-7" />
   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/vApp/vapp-7/metadata" />
</Metadata>
```

3 Modify the retrieved `Metadata` element.

You can add new `MetadataEntry` elements or modify existing ones. If you modify existing ones, your modifications are merged into the object's `Metadata` following the rules listed in VMware Cloud Director API Object Metadata Contents.

4 POST the `Metadata` element to the `rel="add"` link described in Step 2.

See Update a Metadata Element.

Example: Update a Metadata Element

This example updates the empty `Metadata` element shown in Step 2 to create two `MetadataEntry` elements.

In this example, the `Metadata` element contains `MetadataEntry` elements for which the `Domain` is `SYSTEM`. Only the system administrator can update these elements.

Request:

```
POST https://vcloud.example.com/api/admin/vApp/vapp-7/metadata
Content-Type: application/vnd.vmware.vcloud.metadata+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Metadata
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   type="application/vnd.vmware.vcloud.metadata+xml">
   <MetadataEntry
      type="application/vnd.vmware.vcloud.metadata.value+xml">
      <Domain
         visibility="READONLY">SYSTEM</Domain>
      <Key>Organization Web Page</Key>
      <TypedValue
         xsi:type="MetadataStringValue">
         <Value>http://internal.example.com/orgs/Finance</Value>
      </TypedValue>
   </MetadataEntry>
   <MetadataEntry
      type="application/vnd.vmware.vcloud.metadata.value+xml">
      <Domain
         visibility="PRIVATE">SYSTEM</Domain>
      <Key>LOS</Key>
      <TypedValue
         xsi:type="MetadataStringValue">
         <Value>bronze</Value>
      </TypedValue>
   </MetadataEntry>
</Metadata>
```

The response is a `Task`.

Response:

```
<Task
   xmlns="http://www.vmware.com/vcloud/v1.5"
   status="running"
   ...
   operationName="metadataUpdate"
   operation="Updating metadata for vApp (7)"
   ... >
   ...
</Task>
```

After the task is complete, the `Metadata` element is updated to contain the entries specified in the request, along with links that you can use to retrieve or update individual `MetadataEntry` elements.

```
GET https://vcloud.example.com/api/admin/vApp/vapp-7/metadata
...
<Metadata
   xmlns="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.vcloud.metadata+xml"
   href="https://vcloud.example.com/api/vApp/vapp-7/metadata"
   ... >
   <Link
      rel="add"
      type="application/vnd.vmware.vcloud.metadata+xml"
      href="https://vcloud.example.com/api/vApp/vapp-7/metadata" />
   <Link
      rel="up"
      type="application/vnd.vmware.vcloud.vApp+xml"
      href="https://vcloud.example.com/api/vApp/vapp-7" />
   <MetadataEntry>
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.metadata.value+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/metadata/SYSTEM/
Organization+Web+Page" />
      <Link
         rel="remove"
         type="application/vnd.vmware.vcloud.metadata.value+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/metadata/SYSTEM/
Organization+Web+Page" />
      <Link
         rel="up"
         type="application/vnd.vmware.vcloud.metadata+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/metadata" />
      <Domain
         visibility="READONLY">SYSTEM</Domain>
      <Key>Organization Web Page</Key>
      <TypedValue
         xsi:type="MetadataStringValue">
         <Value>http://internal.example.com/orgs/Finance</Value>
      </TypedValue>
   </MetadataEntry>
   <MetadataEntry>
      <Link
         rel="edit"
         type="application/vnd.vmware.vcloud.metadata.value+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/metadata/SYSTEM/LOS" />
      <Link
         rel="remove"
         type="application/vnd.vmware.vcloud.metadata.value+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/metadata/SYSTEM/LOS" />
      <Link
         rel="up"
         type="application/vnd.vmware.vcloud.metadata+xml"
         href="https://vcloud.example.com/api/vApp/vapp-7/metadata" />
      <Domain
         visibility="PRIVATE">SYSTEM</Domain>
      <Key>LOS</Key>
      <TypedValue
         xsi:type="MetadataStringValue">
         <Value>bronze</Value>
      </TypedValue>
   </MetadataEntry>
</Metadata>
```

Retrieve or Update a Metadata Value

Each *name=value* pair in an object's metadata is represented as a MetadataEntry element, which includes links that an administrator or the object owner can use to retrieve or update the metadata value, or delete the MetadataEntry.

Prerequisites

- Verify that you are logged in to the VMware Cloud Director API as an administrator or the object owner.
- Retrieve the object's Metadata element. See Retrieve or Update a Metadata Element

Procedure

1 Examine the retrieved Metadata element to find the MetadataEntry you want.

Each MetadataEntry contains a link of the following form, which you can use when updating the Value of the entry:

```
<Link
   rel="edit"
   type="application/vnd.vmware.vcloud.metadata.value+xml"
   href="https://vcloud.example.com/api/admin/object/id/metadata/DOMAIN/key-name" />
```

For example:

```
<Link
   ...
   href="https://vcloud.example.com/api/admin/org/26/metadata/SYSTEM/LOS/>
```

2 Retrieve or update the value.

Make a request to the URL in the value of the href attribute of the MetadataEntry that contains the Key.

- To retrieve the value, make a GET request to URL in the value of the href attribute of the MetadataEntry. The response is a MetadataValue element.
- To update the value, make a PUT request to the URL in the edit link described in Step 1, and supply a MetadataValue element as the request body. See .Update a Metadata Value

Note The href value of a MetadataEntry in the SYSTEM domain includes the SYSTEM qualifier. The href value of a MetadataEntry in the GENERAL domain does not include a qualifier.

Example: Update a Metadata Value

This request updates the value of the metadata Key named LOS from the original value of bronze (shown in Update a Metadata Element) to a new value of silver. Note that because this MetadataValue is contained by a MetadataEntry where the Domain is SYSTEM and visibility is PRIVATE, only a system administrator can update it.

Request:

```
PUT https://vcloud.example.com/api/admin/org/26/metadata/SYSTEM/LOS
Content-Type: application/vnd.vmware.vcloud.metadata.value+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<MetadataValue
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <TypedValue
      xsi:type="MetadataStringValue">
      <Value>silver</Value>
   </TypedValue>
</MetadataValue>
```

The response is a task.

Response:

```
<Task
   xmlns="http://www.vmware.com/vcloud/v1.5"
   status="running"
   ...
   operationName="metadataUpdate"
   operation="Updating metadata for vApp (7)"
   ... >
   ...
</Task>
```

Using the Query Service

10

You can use the VMware Cloud Director API query service to query the VMware Cloud Director database for information about objects in the cloud.

The query service provides the following kinds of queries:

- Typed queries, which require you to construct a query URL that specifies a query type and optional parameters.
- Packaged queries, which have well-known URLs and can accept many of the same parameters used with typed queries.

Both typed and packaged queries allow you to specify one of the following formats in which to display the result set:

- A `records` format that returns *name*=*value* pairs for all properties of each matching object. This is the default. A *name* can be either a static or dynamic. Static names are predefined object property names, and are returned in an object-specific `QueryRecordResultType` element. For a list of these names, see the `Queries` reference pages in the *VMware Cloud Director API Schema Reference*. Dynamic names are user-defined metadata key names. Dynamic *name*=*value* pairs are not returned unless the object includes a non-empty `Metadata` element. See Chapter 9 Working with Object Metadata.
- An `idrecords` format that is identical to the `records` format, except that object reference *value*s are returned in `id` format rather than `href` format. See Objects, References, and Representations.
- A `references` format that returns a reference in `href` format to each object that matches the query criteria.

Query results are paginated, and include links to previous and next pages where needed. Page size can be specified in the query request. Default and maximum page sizes are specified in the VMware Cloud Director configuration. You can also apply filter criteria to the list of items returned.

Read the following topics next:

- Typed Queries
- Packaged Queries
- Query Parameters

- Add a Metadata Filter to a Query

Typed Queries

Typed queries require you to construct a request URL that specifies an object type and optional parameters. Use this URL with a GET request to return query results.

Query Syntax

Typed queries have the following syntax:

```
API-URL/query?type=name[&param][&param ... ][&filter]
```

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *name* is the name of a query type. Type names are case-sensitive.
- *param* is an optional query parameter. Zero or more parameters are allowed. See Query Parameters.
- *filter* is an optional filter expression. At most one filter expression is allowed. See Filter Expressions.

Query Types

Each query type returns its result set as an XML document in which objects are represented as elements and object properties are represented as attributes, pairing the name of the property with its value at the time the request was made. By default, result sets are returned in the `records` format, which shows all database records for each object. Most queries also support the `references` format, which returns a set of object references, including `name`, `type`, and `href` attributes. All queries that support the `records` format also support the `idrecords` format. For more information about the `format` parameter, see Query Parameters.

Query Reference

The VMware Cloud Director API schema reference includes reference material for all queries. The schema reference is available at https://code.vmware.com.

Example: Listing All Queries

You can retrieve a summary list of all typed queries types accessible to the currently authenticated user by making a request like this one:

```
GET https://vcloud.example.com/api/query
```

The response is a `QueryList` element that contains a `Link` for every query. The `href` value of each `Link` is a URL you can GET to run the query specifying the default format.

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryList
```

```
   type="application/vnd.vmware.vcloud.query.queryList+xml"
   href="https://vcloud.example.com/api/query"
   ... >
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.references+xml"
      name="organization"
      href="https://vcloud.example.com/api/query?type=organization&amp;format=references" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.records+xml"
      name="organization"
      href="https://vcloud.example.com/api/query?type=organization&amp;format=records" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.idrecords+xml"
      name="organization"
      href="https://vcloud.example.com/api/query?type=organization&amp;format=idrecords" />
   <Link
      rel="down"
      type="application/vnd.vmware.vcloud.query.references+xml"
      name="adminOrgNetwork"
      href="https://vcloud.example.com/api/query?
type=adminOrgNetwork&amp;format=references" />
   ...
   </QueryList>
```

If you make a query whose result set you do not have rights to view, a response code of ACCESS_TO_RESOURCE_IS_FORBIDDEN (403) is returned.

Example: Simple Typed Query

This simple typed query retrieves a list of all users in your organization and returns a response in the default (records) format.

```
GET https://vcloud.example.com/api/query?type=user
```

Response:

```
<QueryResultRecords
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="3"
   pageSize="25"
   page="1"
   name="user"
   type="application/vnd.vmware.vcloud.query.records+xml"
   href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=records"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5 http://vcloud.example.com/api/v1.5/
schema/master.xsd">
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.references+xml"
      href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=references"/>
   <UserRecord
      deployedVMQuota="0"
      fullName="User One"
      identityProviderType="INTEGRATED"
      isEnabled="true"
      isLdapUser="false"
      name="bob"
      numberOfDeployedVMs="0"
      numberOfStoredVMs="0"
      storedVMQuota="0"
      href="https://vcloud.example.com/api/admin/user/39"
      storedVMQuotaRank="-1"
      deployedVMQuotaRank="-1"/>
   <UserRecord
      deployedVMQuota="0"
      fullName="User Two"
      identityProviderType="INTEGRATED"
      isEnabled="true"
      isLdapUser="false"
      name="zorro"
      numberOfDeployedVMs="0"
      numberOfStoredVMs="0"
      storedVMQuota="0"
      href="https://vcloud.example.com/api/admin/user/24"
      storedVMQuotaRank="-1"
      deployedVMQuotaRank="-1"/>
   <UserRecord
      deployedVMQuota="0"
      fullName="Example User"
      identityProviderType="INTEGRATED"
      isEnabled="true"
      isLdapUser="false"
      name="nobody"
      numberOfDeployedVMs="0"
      numberOfStoredVMs="0"
      storedVMQuota="0"
      href="https://vcloud.example.com/api/admin/user/58"
      storedVMQuotaRank="-1"
      deployedVMQuotaRank="-1"/>
</QueryResultRecords>
```

This simple typed query retrieves the same list of all users in your organization and returns a response in the references format.

```
GET https://vcloud.example.com/api/query?type=user&format=references
```

Response:

```
<UserReferences
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="3"
   pageSize="25"
   page="1"
   name="user"
   type="application/vnd.vmware.vcloud.query.references+xml"
   href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=references"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ...
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=records" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="bob"
      href="https://vcloud.example.com/api/admin/user/39" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="zorro"
      href="https://vcloud.example.com/api/admin/user/24" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="nobody"
      href="https://vcloud.example.com/api/admin/user/58" />
</UserReferences>
```

Example: Query With Parameters

This query retrieves a list of all users in your organization and returns a response in the records format. The query includes a `sortAsc=name` parameter, so the result set is sorted by object name.

```
GET https://vcloud.example.com/api/query?type=user&format=references&sortAsc=name
```

Response:

```
<UserReferences
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="3"
   pageSize="25"
   page="1"
   name="user"
   type="application/vnd.vmware.vcloud.query.references+xml"
   href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=references"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/query?
type=user&amp;page=1&amp;pageSize=25&amp;format=records" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="bob"
      href="https://vcloud.example.com/api/admin/user/39" />
   <UserReference
```

```
      type="application/vnd.vmware.admin.user+xml"
      name="nobody"
      href="https://vcloud.example.com/api/admin/user/58" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="zorro"
      href="https://vcloud.example.com/api/admin/user/24" />
</UserReferences>
```

This query retrieves a list of all users in your organization and returns a response in the records format. The query includes a `filter=ldapUser==true` parameter, so the result set lists the subset of users who are imported from LDAP. Note that you can filter on a record (attribute) value even though you specify the references format.

```
GET https://vcloud.example.com/api/query?
type=adminUser&format=references&filter=ldapUser==true
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<UserReferences
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="2"
   pageSize="25"
   page="1"
   name="user"
   type="application/vnd.vmware.vcloud.query.references+xml"
   href="https://vcloud.example.com/api/query?
type=adminUser&amp;page=1&amp;pageSize=25&amp;format=references&amp;filter=ldapUser==true"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/
querytype=adminUser&amp;page=1&amp;pageSize=25&amp;format=records&amp;filter=ldapUser==true" /
>
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="bob"
      href="https://vcloud.example.com/api/admin/user/39" />
   <UserReference
      type="application/vnd.vmware.admin.user+xml"
      name="zorro"
      href="https://vcloud.example.com/api/admin/user/58" />
</UserReferences>
```

Packaged Queries

Packaged queries have well-known URLs and can accept most of the parameters used with typed queries.

Query Syntax

Packaged queries have the following syntax:

```
API-URL/query-url[?param][&param ... ][&filter]
```

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *query-url* is the packaged query URL.
- *param* is an optional query parameter. Zero or more parameters are allowed. See Query Parameters.
- *filter* is an optional filter expression. At most one filter expression is allowed. See Filter Expressions.

The response is a `QueryList` element that contains a `Link` for every query. The `href` value of each `Link` is a URL you can GET to run the query specifying the default format.

Each query type returns its result set as an XML document in which objects are represented as elements and object properties are represented as attributes, pairing the name of the property with its value at the time the request was made. By default, result sets are returned in the `records` format, which shows all database records for each object. Most queries also support the `references` format, which returns a set of object references, including `name`, `type`, and `href` attributes. All queries that support the `records` format also support the `idrecords` format. For more information about the `format` parameter, see Query Parameters.

Query Categories

Packaged queries are divided into the following categories:

User queries

The queries have the form *API-URL*/*object-type*/query. Any user can run these queries.

Administrator queries

The queries have the form *API-URL*/admin/*object-type*/query. An organization administrator can run these queries.

Extension queries

The queries have the form *API-URL*/admin/extension/*object-type*/query. A system administrator can run these queries.

Table 10-1. Packaged Queries

Query URL	Result Set
API-URL/catalogs/query	All catalogs in your organization that you have rights to view or modify
API-URL/mediaList/query	All media that you can view or modify

Table 10-1. Packaged Queries (continued)

Query URL	Result Set
API-URL/vAppTemplates/query	All vApp templates that you can view or modify
API-URL/vApps/query	All vApps that you can view or modify
API-URL/vms/query	All virtual machines that you can view or modify
API-URL/admin/groups/query	Groups in all organizations in the system
API-URL/admin/users/query	Users in all organizations in the system
API-URL/admin/strandedUsers/query	Stranded users in the organization. When you delete an LDAP group, users who were imported from that group become stranded and cannot log in.
API-URL/admin/roles/query	All roles defined in the system
API-URL/admin/rights/query	All rights defined in the system
API-URL/admin/orgs/query	All organizations visible to you
API-URL/admin/vdcs/query	All VDCs in the system
API-URL/admin/extension/hostReferences/query	All hosts registered to the system
API-URL/admin/extension/datastores/query	All datastores in the system
API-URL/admin/extension/externalNetworkReferences/query	External networks in the system
API-URL/admin/extension/networkPoolReferences/query	All network pools in the system
API-URL/admin/extension/providerVdcReferences/query	All provider VDCs in the system
API-URL/admin/extension/vimServerReferences/query	All vCenter servers registered to the system
API-URL/admin/extension/orgNetworks/query	All organization networks in the system
API-URL/admin/extension/vapps/query	All vApps in the system
API-URL/admin/extension/orgVdcs/query	All VDCs in the system

Examples

Simple packaged query using the records format, which is the default.

```
GET https://vcloud.example.com/api/catalogs/query
```

Response:

```
<QueryResultRecords
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="15"
   pageSize="25"
   page="1"
   name="catalog"
   type="application/vnd.vmware.vcloud.query.records+xml"
   href="https://vcloud.example.com/api/catalogs/query?
page=1&amp;pageSize=25&amp;format=records"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.references+xml"
      href="https://vcloud.example.com/api/catalogs/query?
type=catalog&amp;page=1&amp;pageSize=25&amp;format=references" />
   <CatalogRecord
      ownerName="system"
      organizationName="VMware"
      numberOfTemplates="30"
      numberOfMedia="3"
      name="VAM"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-21T14:28:09.273-07:00"
      href="https://vcloud.example.com/api/catalog/150" />
   <CatalogRecord
      ownerName="system"
      organizationName="QA"
      numberOfTemplates="0"
      numberOfMedia="1"
      name="QA-Cat"
      isShared="false"
      isPublished="true"
      description=""
      createdOn="2011-03-24T16:37:11.130-07:00"
      href="https://vcloud.example.com/api/catalog/78" />
   <CatalogRecord
      ownerName="system"
      organizationName="Org-d5443f6b-85e"
      numberOfTemplates="0"
      numberOfMedia="1"
      name="Catalog-3f79780c-6b0"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-25T11:27:56.063-07:00"
      href="https://vcloud.example.com/api/catalog/99" />
   <CatalogRecord
      ownerName="system"
      organizationName="Engineering"
      numberOfTemplates="2"
      numberOfMedia="4"
      name="TestCat"
      isShared="true"
      isPublished="true"
      description="New Catalog"
      createdOn="2011-03-22T17:10:10.067-07:00"
      href="https://vcloud.example.com/api/catalog/43" />
   <CatalogRecord
```

```
      ownerName="system"
      organizationName="Engineering"
      numberOfTemplates="8"
      numberOfMedia="1"
      name="catalog1"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-22T16:14:06.360-07:00"
      href="https://vcloud.example.com/api/catalog/91" />
</QueryResultRecords>
```

Packaged query using references format.

```
GET https://vcloud.example.com/api/catalogs/query?format=references
```

Response:

```
<CatalogReferences
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="15"
   pageSize="25"
   page="1"
   name="catalog"
   type="application/vnd.vmware.vcloud.query.references+xml"
   href="https://vcloud.example.com/api/catalogs/query?
page=1&amp;pageSize=25&amp;format=references"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/catalogs/query?
type=catalog&amp;page=1&amp;pageSize=25&amp;format=records" />
    <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="VAM"
      href="https://vcloud.example.com/api/catalog/150" />
   <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="QA-Cat"
      href="https://vcloud.example.com/api/catalog/78" />
   <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="Catalog-3f79780c-6b0"
      href="https://vcloud.example.com/api/catalog/99" />
   <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="TestCat"
      href="https://vcloud.example.com/api/catalog/43" />
   <CatalogReference
      type="application/vnd.vmware.vcloud.catalog+xml"
      name="catalog1"
      href="https://vcloud.example.com/api/catalog/91" />
</CatalogReferences>
```

Packaged query with sorting and filtering. This query adds parameters and a filter expression to produce a list of catalogs that contain one or more vApp templates. The query sorts the result set in ascending order by number of vApp templates:

```
GET https://vcloud.example.com/api/catalogs/query?
format=records&sortAsc=numberOfTemplates&filter=numberOfTemplates!=0
```

Response:

```
<QueryResultRecords
   xmlns="http://www.vmware.com/vcloud/v1.5"
   total="3"
   pageSize="25"
   page="1"
   name="catalog"
   type="application/vnd.vmware.vcloud.query.records+xml"
   href="https://vcloud.example.com/api/
catalogs/query?page=1&amp;pageSize=25&amp;format=records&amp;filter=numberOfTemplates!
=0&amp;sortAsc=numberOfTemplates"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.query.references+xml"
      href="https://vcloud.example.com/api/catalogs/query?
type=catalog&amp;page=1&amp;pageSize=25&amp;format=references&amp;filter=numberOfTemplates!
=0&amp;sortAsc=numberOfTemplates" />
   <CatalogRecord
      ownerName="system"
      organizationName="Engineering"
      numberOfTemplates="2"
      numberOfMedia="4"
      name="TestCatalog"
      isShared="true"
      isPublished="true"
      description="New Catalog"
      createdOn="2011-03-22T17:10:10.067-07:00"
      href="https://vcloud.example.com/api/catalog/43" />
   <CatalogRecord
      ownerName="system"
      organizationName="Engineering"
      numberOfTemplates="8"
      numberOfMedia="1"
      name="catalog1"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-22T16:14:06.360-07:00"
      href="https://vcloud.example.com/api/catalog/91" />
   <CatalogRecord
      ownerName="system"
      organizationName="VMware"
      numberOfTemplates="30"
      numberOfMedia="3"
      name="VAM"
      isShared="true"
      isPublished="true"
      description=""
      createdOn="2011-03-21T14:28:09.273-07:00"
      href="https://vcloud.example.com/api/catalog/150" />
</QueryResultRecords>
```

Query Parameters

Query parameters specify result set properties such as pagination, sort order, and filter criteria.

Query Parameters

Entered queries must include a `type` parameter, which specifies the type of query to run. Packaged queries cannot specify a type parameter. All other parameters are optional. If a parameter is omitted, a default value is assumed.

Table 10-2. Query Parameters

Parameter Name	Parameter Description	Default
`fields`	Comma-separated list of attribute names or metadata key names to return. For example, `fields=name,isEnabled` or `fields=metadata:rank`. See Specifying Metadata in a Query or a Filter Expression.	Returns all static attribute names. Returns metadata only if the object has a non-empty `Metadata` element.
`filter`	Filter expression. The filter expression must always be encoded to comply with RFC3986 (pct-encoded). If your filter values contain special characters, such as ( ) , ; \, you must escape the special characters with a backslash (\). For example, in `filter=name==VM\,1`, the name to search for is `VM,1`, and in `filter=name==VM\,%201`, the name to search for is `VM, 1` because it has both encoded and escaped characters. See Table 10-3. Query Filter Expressions and Encoding Filter Expressions.	None
`format`	One of the following types: **`references`** Returns a reference to each object, including its `name`, `type`, and `href` attributes. **`records`** Returns all database records for each object, with each record as an attribute. **`idrecords`** Identical to the `records` format, except that object references are returned in `id` format rather than `href` format.	`format=records`
`links`	Boolean value specifying whether to include `Link` elements in the result set for certain object types.	`links=false`

Table 10-2. Query Parameters (continued)

Parameter Name	Parameter Description	Default
`offset`	Integer value specifying the first record to return. Record numbers < `offset` are not returned.	`offset=0`
`page`	If the query results span multiple pages, return this page.	`page=1`
`pageSize`	Number of results per page, to a maximum of 128.	`pageSize=25`
`sortAsc`=*attribute-name*	Sort results by *attribute-name* in ascending order. *attribute-name* cannot include metadata. For information about the attributes that can be used with `sortAsc`, see Attribute Names and Values.	Sorted by database ID
`sortDesc`=*attribute-name*	Sort results by *attribute-name* in descending order. *attribute-name* cannot include metadata. For information about the attributes that can be used with `sortDesc`, see Attribute Names and Values.	Sorted by database ID
`type`	The type of the query. Type names are case-sensitive. See Query Types.	None. This parameter is required for all typed queries, and is not allowed for any packaged query.

Filter Expressions

For queries that do not examine object metadata, you can filter results using string matching or numeric comparison operations. A filter comprises one or more subexpressions drawn from the following set of operators.

Table 10-3. Query Filter Expressions

Operator	Example	Operation
`==`	*attribute*==*value*	Matches. The example evaluates to true if *attribute* has a value that matches *value* in a case-sensitive comparison. **Note** Asterisk (*) characters that appear anywhere in *value* are treated as wildcards that match any character string. When *value* includes wildcards, the comparison with *attribute* becomes case-insensitive.
`!=`	*attribute*!=*value*	Does not match. The example evaluates to true if *attribute* has a value that does not match *value* in a case-sensitive comparison. Wildcard characters are not allowed. You must include `filterEncoded=true` when adding a filter that includes an inequality check.
`;`	*attribute1*==*value1*; *attribute2*!=*value2*	Logical AND. The example evaluates to true only if *attribute1* has a value that matches *value1* and *attribute2* has a value that does not match *value2* in a case-sensitive comparison.

Table 10-3. Query Filter Expressions (continued)

Operator	Example	Operation
,	*attribute1*==*value1*, *attribute2*==*value2*	Logical OR. The example evaluates to true if *attribute1* has a value that matches *value1* or *attribute2* has a value that matches *value2* in a case-sensitive comparison.
`=gt=`	*attribute*`=gt=`*value*	Greater than. The example evaluates to true if *attribute* has a value that is greater than *value*. Both *attribute* and *value* must be of type `int`, `long`, or `dateTime`.
`=lt=`	*attribute*`=lt=`*value*	Less than. The example evaluates to true if *attribute* has a value that is less than *value*. Both *attribute* and *value* must be of type `int`, `long`, or `dateTime`.
`=ge=`	*attribute*`=ge=`*value*	Greater than or equal to. The example evaluates to true if *attribute* has a value that is greater than or equal to *value*. Both *attribute* and *value* must be of type `int`, `long`, or `dateTime`.
`=le=`	*attribute*`=le=`*value*	Less than or equal to. The example evaluates to true if *attribute* has a value that is less than or equal to *value*. Both *attribute* and *value* must be of type `int`, `long`, or `dateTime`.

Not all attributes can be used in a filter expression. For details, see the reference pages for query result types in the *VMware Cloud Director API Schema Reference*.

Specifying Metadata in a Query or a Filter Expression

Because metadata values are dynamic and metadata names have an optional domain qualifier queries that filter metadata must specify a qualified name, a value, and the type of the value.

- The domain must be specified for any `MetadataEntry` that is in the `SYSTEM` domain. If no DOMAIN is specified, the query returns the value for key name in the `GENERAL` domain, if it exists.
- The type of the value must be specified, using one of the following keywords.

  Table 10-4. Metadata Type Specifiers in Query Filters

Type Name as Specified in TypedValue	Type Name as a Filter Expression Keyword
MetadataStringValue	STRING
MetadataNumberValue	NUMBER
MetadataDateTimeValue	DATETIME
MetadataBooleanValue	BOOLEAN

For queries that examine object metadata, you can filter query results using numeric comparison operations when a metadata value has type NUMBER or DATETIME. Because object metadata types are dynamic, filter expressions for metadata queries require additional parameters that define the *attribute* part of the expression as metadata, and specify the type of the *value* part of the expression.

Table 10-5. Metadata Query Filter Expressions

Operator	Example	Operation
`=gt=`	`metadata:`*attribute*`=gt=[NUMBER | DATETIME]:`*value*	Greater than. The example evaluates to true if *value* is of type NUMBER or DATETIME and the value of the metadata key named *attribute* is greater than *value*.
`=lt=`	`metadata:`*attribute*`=lt=[NUMBER | DATETIME]:`*value*	Less than. The example evaluates to true if *value* is of type NUMBER or DATETIME and the value of the metadata key named *attribute* is less than *value*.
`=ge=`	`metadata:`*attribute*`=ge=[NUMBER | DATETIME]:`*value*	Greater than or equal to. The example evaluates to true if *value* is of type NUMBER or DATETIME and the value of the metadata key named *attribute* is greater than or equal to *value*.
`=le=`	`metadata:`*attribute*`=le=[NUMBER | DATETIME]:`*value*	Less than or equal to. The example evaluates to true if *value* is of type NUMBER or DATETIME and the value of the metadata key named *attribute* is less than or equal to *value*.

You can specify up to 8 metadata fields in a single query. You cannot use a metadata value as a sort key when sorting query output.

The syntax for specifying metadata in a `fields` parameter is:

```
fields=metadata[@SYSTEM]:key-name
```

For example:

```
fields=metadata:rank
```

```
fields=metadata@SYSTEM:expiry
```

The syntax for specifying a metadata field in a filter expression is:

```
metadata[@SYSTEM]:key-name operator type-keyword:value
```

For example:

```
metadata:rank=ge=NUMBER:1
```

```
metadata@SYSTEM:expiry=le=DATETIME:2012-06-18T12:00:00-05:00
```

Encoding Filter Expressions

When you are comparing a *key* with a literal *value* that includes characters that might be subject to interpretation when used in a URL (often termed unsafe characters), the *value* must be encoded as described in RFC3986 (pct-encoded). For example, to create a filter expression to match a `hostName` property whose value is `12&345`, encode the value as shown here:

```
filter=hostName==12%26345
```

When a filter expression includes metadata, you must encode both the *key* and the *value* this way. In queries, you must encode the `%` symbol as `%25`. For example, this query, which includes metadata, would require this sort of additional encoding.

```
https://vcloud.example.com/api/query?
type=organization&format=records
&fields=metadata:rank,metadata@SYSTEM:expiry
&filter=metadata%40SYSTEM%3Aexpiry%3Dlt%3DDATETIME%3ADATETIME%253A2012-05-01T00%253A00%253A00.
000-04%253A00DATETIME%253A2012-05-01T00%253A00%253A00.000-04%253A00
```

Grouping Filter Subexpressions

Group filter subexpressions with parentheses. Separate grouped subexpressions with a semicolon (no spaces).

```
(attribute1==value1;attribute2!=value2);(attribute3==value3;attribute4!=value4)...
```

For example:

```
https://vcloud.example.com/api/query?
type=providerVdcResourcePoolRelation&format=records&filter=(numberOfVMs!=0;isPrimary==true)
```

Attribute Names and Values

Several parameters and all filter expressions require you to specify an attribute name. To understand the attribute names included in a particular result set, you can use the *VMware Cloud Director API Schema Reference* available at https://code.vmware.com.

1 In the schema reference, find your query, then open the query reference page.

2 In the **Record Result** section, click the link to open a page that shows the name, type, description, and other information for each attribute returned by the query.

   Attributes that can be used in a filter expression have a YES in the FILTER column.

   Attributes that can be used with the `sortAsc` or `sortDesc` parameter have a YES in the SORT column.

Add a Metadata Filter to a Query

All packaged queries and most typed queries return object metadata if it exists for an object in the result set. You can add metadata-specific filter criteria to a query.

To query an object's metadata, add a metadata field specifier to the query. This specifier follows the `fields` parameter and consists of the string `metadata:` followed by the name of a metadata key.

Note You must use a `metadata@SYSTEM` parameter to specify metadata keys in the `SYSTEM` domain. Metadata in the `GENERAL` domain does not require a qualifier. Up to eight `metadata` parameters can be included in a query.

Procedure

1 Create the filter expression.

2 Add a `fields` parameter that specifies `metadata`.

The syntax for specifying a metadata field is:

```
fields=metadata:key-name [, metadata:key-name ] ...
```

3 Add a filter expression to the query.

All of the standard filter expressions are supported for metadata queries. In addition, a set of numeric comparisons are available for values of type NUMBER or DATETIME.

Example: A Query With a Metadata Filter

Assume that the vApps in your organization have metadata like that shown in Update a Metadata Element. To retrieve a list of all the vApps that have a key named `PenTested` whose value is `false`, use a query and filter like this one.

Request:

```
GET https://vcloud.example.com/api/query?type=vApp&format=records
&fields=metadata:PenTested&filter="metadata:PenTested==false"
```

Note When a filter expression includes metadata, you must encode both the *key* and the *value* as described in RFC3986. See Encoding Filter Expressions.

Configuring and Using Blocking Tasks and Notifications

11

VMware Cloud Director allows a system administrator to configure many operations as blocking tasks, which are suspended until a system administrator acts on them or a preconfigured timer expires. Blocking tasks generate AMQP messages that you can use to automate the handling of the underlying user request. A system administrator can also enable nonblocking AMQP notifications of all system events.

The system administrator can configure the VMware Cloud Director AMQP service to provide a stream of notifications about events in the cloud. By configuring specific tasks as blocking and writing AMQP clients that process the messages generated when these tasks are launched, you can create a programmatic facility for reviewing and acting on tenant requests. When a user requests an operation that has been configured as a blocking task, the system sends a message about the task to the configured AMQP broker. The system also creates a reference to the task in the cloud's `BlockingTaskReferences` container. A system administrator can retrieve the list of `BlockingTask` elements by making a GET request to the system's `blockingTasks` link, or to a URL included in the AMQP message.

For more information about the VMware Cloud Director AMQP service, see #unique_231.

Subscribing to Notifications

Notifications of system events are sent to the AMQP message broker that was configured in the system AMQP settings. Notifications are always generated in two formats:

- An XML document, which is sent to the AMQP exchange specified in the system `AmqpSettings`.
- A JSON object, which is sent to an AMQP exchange whose name has the form *`prefix`*`.notifications20`, where *prefix* is the value of the `AmqpPrefix` element in the system `AmqpSettings`.

See Notification Message Format.

AMQP client programs can connect to the broker and specify components of the AMQP routing key to indicate their interest in messages based on content. For example, a client can use the routing key to request the broker to send it all messages from a specific organization, or all messages that indicate a failed task. See Routing Key Format.

Processing Messages from Blocking Tasks

Messages from blocking tasks are also sent to the configured message broker, and clients can use the routing key to indicate their interest in these messages. See Subscribing to Notifications. Messages from blocking tasks contain additional information about the task itself. Clients that process these messages can use the VMware Cloud Director API to authenticate to the system and act on the blocked task.

Read the following topics next:

- Configure Notifications
- Retrieve or Update Blocking Task Settings
- Monitor Blocking Tasks
- Take Action on a Blocking Task
- Extend The Timeout Expiration of an Active Task
- Notification Message Format

Configure Notifications

The system administrator can enable or disable notification messages for events in a cloud.

Notifications are enabled by default. A system administrator can disable them.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the `SystemSettings` element.

2 Examine the response to locate the link that you can use to retrieve the system's `NotificationsSettings` element.

  This link has a `rel` attribute value of `down` and a `type` attribute value of `application/vnd.vmware.admin.notificationsSettings+xml`, as shown here:

```
<Link href="https://vcloud.example.com/admin/extension/settings/notifications"
   rel="down"
   type="application/vnd.vmware.admin.notificationsSettings+xml"/>
```

3 Enable or disable notifications.

   a Retrieve the NotificationsSettings element.

      Make a GET request to the href value of the application/vnd.vmware.admin.notificationsSettings+xml link.

   b Modify the value of the EnableNotifications element to enable or disable notifications.

   c Update the modified element with the new contents.

      PUT the modified element to the href value of its rel="edit" link.

Example: Enable Notifications

This request sets the value of EnableNotifications to true. This is the default value, so you normally would not need to make this request unless notifications had been disabled by a previous request.

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<vmext:NotificationsSettings
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.notificationsSettings+xml">
   <vmext:EnableNotifications>true</vmext:EnableNotifications>
</vmext:NotificationsSettings>
```

The response contains information extracted from the request, and adds an edit link that you can use to change the value of this element.

Response:

```
200 OK
...
<vmext:NotificationsSettings
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   href="https://vcloud.example.com/api/admin/extension/settings/notifications"
   type="application/vnd.vmware.admin.notificationsSettings+xml">
   <vcloud:Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/extension/settings/notifications"
      type="application/vnd.vmware.admin.notificationsSettings+xml" />
   <vmext:EnableNotifications>true</vmext:EnableNotifications>
</vmext:NotificationsSettings>
```

Retrieve or Update Blocking Task Settings

Timeout settings, default actions, and related messages for blocking tasks are properties of a cloud. They apply to all organizations in the cloud. Only a system administrator can view or modify them.

When a user requests an operation that is configured to create a blocking task, the system creates a reference to the operation in the cloud's `BlockingTaskReferences` container. The system also sends a message about the task to the configured AMQP broker. A system administrator can retrieve the list of `BlockingTaskReferences` by making a GET request to the system's `blockingTasks` link. An AMQP client can use information in the message to construct a URL that it can use to retrieve the task. See Notification Message Format.

If no action is taken on the blocking task within a specified timeout interval, it is subject to a default action. You can specify the timeout interval and default action for all blocking tasks by modifying the system's `BlockingTaskSettings` element. To configure an operation as a blocking task, add the operation name to the `BlockingTaskOperations` element contained by `BlockingTaskSettings`. See Task Operations.

Prerequisites

- This operation is restricted to system administrators.
- Retrieve the `SystemSettings` element. See Retrieve or Update System Settings.

Procedure

1 Examine the response to locate the link that you can use to retrieve the system's `BlockingTaskSettings` element.

   This link has a `rel` attribute value of `down` and a `type` attribute value of `application/vnd.vmware.admin.blockingTaskSettings+xml`, as shown here:

```
<Link href="https://vcloud.example.com/api/admin/extension/settings/blockingTask"
   rel="down"
   type="application/vnd.vmware.admin.blockingTaskSettings+xml"/>
```

2 Retrieve the element.

   Make a GET request to the `href` value of the link.

3 (Optional) Modify the element as needed to change the settings it controls.

   See the schema reference.

4 (Optional) Update the modified element with the new contents.

   PUT the modified element to the `href` value of its `rel="edit"` link. See Update Blocking Task Settings.

Example: Update Blocking Task Settings

This request modifies the blocking task settings for a cloud to set the time-out period to 24 hours and adds media upload as an operation that creates a blocking task. See Task Operations for a list of operation names.

Request:

```
PUT https://vcloud.example.com/api/admin/extension/settings/blockingTask
Content-Type: application/vnd.vmware.admin.blockingTaskSettings+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<BlockingTaskSettings
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns="http://www.vmware.com/vcloud/extension/v1.5">
   <TimeoutAction>abort</TimeoutAction>
   <BlockingTaskOperations>
      <vcloud:Operation>vdcUploadMedia</vcloud:Operation>
   </BlockingTaskOperations>
   <TimeoutInMilliseconds>86400000</TimeoutInMilliseconds>
</BlockingTaskSettings>
```

The response contains information extracted from the request, and adds the `href` attributes and `edit` links for the `BlockingTaskSettings` element and the `BlockingTaskOperations` element it contains.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.blockingTaskSettings+xml
...
<BlockingTaskSettings
   xmlns="http://www.vmware.com/vcloud/v1.5"
   TimeoutInMilliseconds="86400000"
   type="application/vnd.vmware.admin.blockingTaskSettings+xmll"
   href="https://vcloud.example.com/api/admin/extension/settings/blockingTask"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ... >
   <Link
      rel="edit"
      type="application/vnd.vmware.admin.blockingTaskSettings+xml"
      href="https://vcloud.example.com/api/admin/extension/settings/blockingTask" />
   <Link
      rel="down"
      type="application/vnd.vmware.admin.taskOperationList+xml"
      href="https://vcloud.example.com/api/admin/extension/settings/blockingTask/operations"/>
   <TimeoutAction>abort</TimeoutAction>
   <BlockingTaskOperations
      type="application/vnd.vmware.admin.taskOperationList+xml"
      href="https://vcloud.example.com/api/admin/extension/settings/blockingTask/operations">
      <Link
         rel="edit"
         type="application/vnd.vmware.admin.taskOperationList+xml"
         href="https://vcloud.example.com/api/admin/extension/settings/blockingTask/
operations"/>
      <Operation>vdcUploadMedia</Operation>
   </BlockingTaskOperations>
</BlockingTaskSettings>
```

Task Operations

Task operation names represent the requests that you can configure as blocking tasks.

To configure a request type as a blocking task, place the operation name in an `Operation` element and add that element to the cloud's `BlockingTaskOperations` element. See Retrieve or Update Blocking Task Settings.

Note These operation names also appear in the `operationName` attribute of the `Task` element that tracks the operation.

Category	Operation Name	Description
EDGE_CLUSTER	edgeClusterConfigure	Configure an edge cluster.
EDGE_CLUSTER	edgeClusterCreate	Create an edge cluster.
EDGE_CLUSTER	edgeClusterDelete	Delete an edge cluster.
EDGE_GATEWAY_SERVICES	bgpConfigUpdate	Configure BGP.
EDGE_GATEWAY_SERVICES	bgpNeighborCreate	Add a BGP neighbor.
EDGE_GATEWAY_SERVICES	bgpNeighborDelete	Delete a BGP neighbor.
EDGE_GATEWAY_SERVICES	bgpNeighborUpdate	Edit a BGP neighbor.
EDGE_GATEWAY_SERVICES	createIpSecVpnTunnel	Create an IPsec VPN tunnel.
EDGE_GATEWAY_SERVICES	createL2VpnTunnel	Create an L2 VPN tunnel.
EDGE_GATEWAY_SERVICES	createNatRule	Create a NAT rule.
EDGE_GATEWAY_SERVICES	deleteFirewallRule	Delete a firewall rule.
EDGE_GATEWAY_SERVICES	deleteFirewallRules	Delete the firewall rules on an edge gateway.
EDGE_GATEWAY_SERVICES	deleteIpSecVpnTunnel	Delete an IPsec VPN tunnel.
EDGE_GATEWAY_SERVICES	deleteL2VpnTunnel	Delete an L2 VPN tunnel.
EDGE_GATEWAY_SERVICES	deleteNatRule	Delete a NAT rule.
EDGE_GATEWAY_SERVICES	gatewayLoadBalancerConfigUpdate	Configure load balancer.
EDGE_GATEWAY_SERVICES	gatewayLoadBalancerPoolCreate	Add a load balancer server pool.
EDGE_GATEWAY_SERVICES	gatewayLoadBalancerPoolDelete	Delete a load balancer server pool.
EDGE_GATEWAY_SERVICES	gatewayLoadBalancerPoolUpdate	Update a load balancer server pool.
EDGE_GATEWAY_SERVICES	gatewayLoadBalancerVirtualServiceCreate	Create a virtual service.
EDGE_GATEWAY_SERVICES	gatewayLoadBalancerVirtualServiceDelete	Delete a virtual service.

Category	Operation Name	Description
EDGE_GATEWAY_SERVICES	gatewayLoadBalancerVirtualServiceUpdate	Update a virtual service.
EDGE_GATEWAY_SERVICES	nsxProxyNatConfigure	Configure NAT.
EDGE_GATEWAY_SERVICES	nsxProxyResourceConfigure	Generic task for configuring NSX Data Center for vSphere resources.
EDGE_GATEWAY_SERVICES	nsxProxyResourceConfigureServices	Generic task for configuring NSX Data Center for vSphere edge services.
EDGE_GATEWAY_SERVICES	orgVdcGatewayDhcpForwarderUpdate	Edit the DHCP forwarding configuration on an edge gateway.
EDGE_GATEWAY_SERVICES	orgVdcGatewayDnsDelete	Delete DNS services.
EDGE_GATEWAY_SERVICES	orgVdcGatewayDnsUpdate	Configure DNS services.
EDGE_GATEWAY_SERVICES	prefixListCreate	Create an IP prefix list.
EDGE_GATEWAY_SERVICES	prefixListDelete	Delete an IP prefix list.
EDGE_GATEWAY_SERVICES	prefixListUpdate	Edit an IP prefix list.
EDGE_GATEWAY_SERVICES	routeAdvertisementUpdate	Configure route advertisement.
EDGE_GATEWAY_SERVICES	slaacProfileUpdate	Configuring SLAAC services on the Edge Gateway.
EDGE_GATEWAY_SERVICES	updateFirewallRule	Edit a firewall rule.
EDGE_GATEWAY_SERVICES	updateFirewallRules	Edit the firewall rules for an edge gateway.
EDGE_GATEWAY_SERVICES	updateIpSecVpnTunnel	Edit an IPSec VPN tunnel.
EDGE_GATEWAY_SERVICES	updateIpSecVpnTunnelProperties	Edit the properties of an IPSec VPN tunnel.
EDGE_GATEWAY_SERVICES	updateL2VpnTunnel	Edit an L2 VPN tunnel.
EDGE_GATEWAY_SERVICES	updateNatRule	Edit a NAT rule.
EDGE_GATEWAY_TASKS	edgeGatewayUpdate	Edit an edge gateway.
EDGE_GATEWAY_TASKS	edgeGatewayUpdateProperties	Edit the properties of an edge gateway.
EDGE_GATEWAY_TASKS	gatewayQosUpdate	Configure QoS Rate Limits on an edge gateway.
EDGE_GATEWAY_TASKS	networkConfigureEdgeGatewaySyslogServerSettings	Configure edge gateway syslog settings.
EDGE_GATEWAY_TASKS	networkEdgeGatewayCreate	Create an edge gateway.

Category	Operation Name	Description
EDGE_GATEWAY_TASKS	networkEdgeGatewayDelete	Delete an edge gateway.
EDGE_GATEWAY_TASKS	networkEdgeGatewayRedeploy	Redeploy edge gateway.
EDGE_GATEWAY_TASKS	networkEdgeGatewaySynchronizeSyslog	Synchronizes the Syslog configuration on the edge gateway.
EDGE_GATEWAY_TASKS	networkEdgeGatewayUpgradeConfiguration	Upgrade an edge gateway.
EDGE_GATEWAY_TASKS	networkGatewayDisableDistributedRouting	Deactivate distributed routing for an edge gateway.
EDGE_GATEWAY_TASKS	networkGatewayEnableDistributedRouting	Activate distributed routing for an edge gateway.
EDGE_GATEWAY_TASKS	networkGatewayFormFactorModify	Modify the form factor of an edge gateway.
EDGE_GATEWAY_TASKS	orgVdcGatewayCreate	Create an organization VDC egde gateway.
EDGE_GATEWAY_TASKS	orgVdcGatewayDelete	Delete an organization VDC edge gateway.
EDGE_GATEWAY_TASKS	orgVdcGatewayUpdate	Edit the settings of an organization VDC edge gateway.
EXTERNAL_NETWORK	externalNetworkCreate	Create an external network.
EXTERNAL_NETWORK	externalNetworkDelete	Delete an external network.
EXTERNAL_NETWORK	externalNetworkUpdate	Update an external network.
EXTERNAL_NETWORK	networkCreateExternalNetwork	Create an external network.
MISCELLANEOUS_TASKS	catalogCachePublishedItem	Cache published catalog item.
MISCELLANEOUS_TASKS	catalogCachePublishedItems	Cache published catalog items.
MISCELLANEOUS_TASKS	catalogDelete	Delete a catalog.
MISCELLANEOUS_TASKS	catalogItemDelete	Delete a catalog item.
MISCELLANEOUS_TASKS	catalogItemEnableDownload	Enable a catalog item for download.
MISCELLANEOUS_TASKS	catalogItemSync	Sync a catalog item.
MISCELLANEOUS_TASKS	catalogSync	Update a subscribed catalog item from its external source.
MISCELLANEOUS_TASKS	catalogSyncAll	Update an externally subscribed catalog from its source.

Category	Operation Name	Description
MISCELLANEOUS_TASKS	importIntoExistingVapp	Import a virtual machine from vCenter Server to an existing vApp.
MISCELLANEOUS_TASKS	importMedia	Import a media object from vSphere.
MISCELLANEOUS_TASKS	importSingletonTemplate	Import a virtual machine from vCenter Server as a vApp template.
MISCELLANEOUS_TASKS	importSingletonVapp	Import a virtual machine from vCenter Server as a vApp.
MISCELLANEOUS_TASKS	importVcVmsIntoExistingVApp	Import a virtual machine from vCenter Server into an existing vApp.
MISCELLANEOUS_TASKS	vdcUploadMedia	Upload media.
MISCELLANEOUS_TASKS	vdcUploadOvfContents	Upload other OVF package contents.
NETWORK	networkDelete	Delete a network.
NETWORK	networkUpdateNetwork	Modify one or more properties of a network object.
NETWORK_POOL	networkCreateNetworkPool	Create a network pool.
NETWORK_POOL	networkDeleteNetworkPool	Delete a network pool.
NETWORK_POOL	networkMergeNetworkPools	Merge network pools.
NETWORK_POOL	networkPoolCreate	Create a network pool.
NETWORK_POOL	networkPoolDelete	Delete a network pool.
NETWORK_POOL	networkPoolUpdate	Update a network pool.
NETWORK_POOL	networkRepairNetworkPool	Repair a network pool.
NETWORK_POOL	networkSyncNetworkPool	Sync a network pool.
NETWORK_POOL	networkUpdateNetworkPool	Modify one or more properties of a VlanPoolType network pool object.
NETWORK_POOL	networkUpdateVlanPool	Modify one or more properties of any network pool object.
ORG_RELATED	orgDeleteUser	Delete a user in an organization.
RESOURCE_CREATION_TASKS	rclCreateProviderVdc	Create a provider VDC.

Category	Operation Name	Description
RESOURCE_CREATION_TASKS	rclDeleteProviderVdc	Delete a provider VDC.
RESOURCE_CREATION_TASKS	rclEnableVxlanForProviderVdc	Enable a VXLAN pool associated with a new provider VDC.
RESOURCE_CREATION_TASKS	rclMergePvdc	Merge provider VDCs.
TENANT_NETWORK	crossVdcNetworkConfigure	Configure cross VDC network.
TENANT_NETWORK	crossVdcNetworkCreate	Create a data center group network.
TENANT_NETWORK	crossVdcNetworkDelete	Delete a data center group network.
TENANT_NETWORK	networkCreateOrgVdcNetwork	Create an organization VDC network.
TENANT_NETWORK	networkDhcpBindingCreate	Create a DHCP binding.
TENANT_NETWORK	networkDhcpBindingDelete	Delete a DHCP binding.
TENANT_NETWORK	networkDhcpBindingUpdate	update a DHCP binding.
TENANT_NETWORK	networkResetOrgVdcNetwork	Reset an organization VDC network.
TENANT_NETWORK	networkSegmentProfilesUpdate	Update the segment profile templates.
TENANT_NETWORK	networkSyncSyslogSettings	Synchronize syslog settings for a network.
TENANT_NETWORK	orgVdcNetworkCreate	Create an organization VDC network.
TENANT_NETWORK	orgVdcNetworkDelete	Delete an organization VDC network.
TENANT_NETWORK	orgVdcNetworkUpdate	Update an organization VDC network.
TKG_TASKS	tkgClusterCreate	Create a Tanzu Kubernetes cluster.
TKG_TASKS	tkgClusterDelete	Delete a Tanzu Kubernetes cluster.
TKG_TASKS	tkgClusterUpdate	Update a Tanzu Kubernetes cluster.
VAPP_TASKS	templateUpdateVm	Update template VM.
VAPP_TASKS	vappAttachDisk	Attach an independent disk to any virtual machine in a vApp.

Category	Operation Name	Description
VAPP_TASKS	vappCheckVmCompliance	Check the storage profile compliance of a virtual machine.
VAPP_TASKS	vappCreateSnapshot	Create a snapshot for the vApp.
VAPP_TASKS	vappDeploy	Deploy a vApp.
VAPP_TASKS	vappDetachDisk	Detach an independent disk from any virtual machine in a vApp.
VAPP_TASKS	vappMigrateVms	Migrate the virtual machines in a vApp to a new host.
VAPP_TASKS	vappPowerOff	Power-off a vApp.
VAPP_TASKS	vappRebootGuest	Reboot any virtual machine in a vApp.
VAPP_TASKS	vappRemoveAllSnapshots	Remove all snapshots for this vApp.
VAPP_TASKS	vappReset	Reset any virtual machine in a vApp.
VAPP_TASKS	vappRevertToCurrentSnapshot	Revert the vApp to its current snapshot.
VAPP_TASKS	vappShutdownGuest	Shut down any virtual machine in a vApp.
VAPP_TASKS	vappSuspend	Suspend any virtual machine in a vApp.
VAPP_TASKS	vappUndeployPowerOff	Undeploy any virtual machine in a vApp by powering it off.
VAPP_TASKS	vappUndeploySuspend	Undeploy any virtual machine in a vApp by suspending it.
VAPP_TASKS	vappUpdateVm	Modify one or more properties of a virtual machine.
VAPP_TASKS	vappUpgradeHwVersion	Upgrade the hardware version of any virtual machine in a vApp.
VDC_GROUP	universalEgressPointCreate	Add an egress point to a data center group.
VDC_GROUP	universalEgressPointDelete	Remove an egress point from a data center group.
VDC_GROUP	universalEgressPointSync	Sync universal egress point.
VDC_GROUP	universalRouterConfigure	Configure universal router.

Category	Operation Name	Description
VDC_GROUP	universalRouterCreate	Create universal router.
VDC_GROUP	universalRouterDelete	Delete universal router.
VDC_GROUP	vdcGroupConfigure	Edit a data center group.
VDC_GROUP	vdcGroupCreate	Create a data center group.
VDC_GROUP	vdcGroupDelete	Delete a data center group.
VDC_GROUP	vdcGroupDfwDisable	Deactivate the distributed firewall service for a data center group.
VDC_GROUP	vdcGroupDfwEnable	Activate the distributed firewall service for a data center group.
VDC_GROUP	vdcGroupDfwPolicyUpdate	Update the single default security policy for a data center group.
VDC_GROUP	vdcGroupDfwRuleDelete	Delete a distributed firewall rule.
VDC_GROUP	vdcGroupDfwRulesUpdate	Update the distributed firewall rules for a data center group.
VDC_GROUP	vdcGroupDfwRuleUpdate	Update a distributed firewall rules for a data center group.
VDC_TASKS	legacyVdcUpdateMedia	Update one or more properties of a media object.
VDC_TASKS	legacyVdcUpdateTemplate	Update one or more properties of a vApp template.
VDC_TASKS	vdcCaptureTemplate	Capture a vApp as a vApp template.
VDC_TASKS	vdcComposeVapp	Compose a vApp.
VDC_TASKS	vdcCopyMedia	Copy a media object.
VDC_TASKS	vdcCopyTemplate	Copy a vApp template.
VDC_TASKS	vdcCopyVapp	Clone a vApp.
VDC_TASKS	vdcCreateComputePolicy	Create a compute policy.
VDC_TASKS	vdcCreateDisk	Create an independent disk.
VDC_TASKS	vdcCreateVdc	Create an organization VDC.
VDC_TASKS	vdcDeleteComputePolicy	Delete a compute policy.
VDC_TASKS	vdcDeleteDisk	Delete an independent disk.

Category	Operation Name	Description
VDC_TASKS	vdcDeleteMedia	Delete a media object.
VDC_TASKS	vdcDeleteTemplate	Delete a vApp template.
VDC_TASKS	vdcDeleteVapp	Delete a vApp.
VDC_TASKS	vdcDeleteVdc	Delete an organization VDC.
VDC_TASKS	vdcEnableDownload	Enable a vApp template for download as OVF.
VDC_TASKS	vdcInstantiateVapp	Instantiate a vApp template.
VDC_TASKS	vdcMoveDisk	Update on a named disk with a new organization VDC.
VDC_TASKS	vdcMoveVapp	Move the vApp's virtual machines to another resource pool, datastore, or network pool.
VDC_TASKS	vdcNetworkProfileConfigure	Configure a network profile.
VDC_TASKS	vdcNetworkProfileDelete	Delete a network profile.
VDC_TASKS	vdcRecomposeVapp	Recompose a vApp.
VDC_TASKS	vdcUpdateComputePolicy	Update a compute policy.
VDC_TASKS	vdcUpdateDisk	Update an independent disk.
VDC_TASKS	vdcUpdateMedia	Update one or more properties of a media object.
VDC_TASKS	vdcUpdateStorageProfiles	Update the storage profiles of a VDC.
VDC_TASKS	vdcUpdateTemplate	Update one or more properties of a vApp template.
VDC_TASKS	vdcUpdateVapp	Update any section of a vApp.
VDC_TASKS	vdcUpdateVdc	Modify one or more properties of an organization VDC.

Monitor Blocking Tasks

A system administrator can retrieve a list of all pending and active blocking tasks. Links in the returned `BlockingTask` element allow the administrator to take action on the request.

In addition to being subject to programmatic action by an AMQP client (see Notification Message Format), blocking tasks can be monitored and managed by a system administrator using the VMware Cloud Director API.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the list of blocking tasks.

  See the request portion of Retrieve a List of Blocking Tasks. If the `BlockingTaskReferences` element contains no `Reference` elements, no blocking tasks are currently active in the system.

2 Retrieve an individual blocking task from one of the `Reference` elements in the response.

  See the request portion of Handling a Blocking Task.

3 Use one of the `action` links in the `BlockingTask` to take action on the task.

  See the response portion of Handling a Blocking Task.

Example: Retrieve a List of Blocking Tasks

Request:

```
GET https://vcloud.example.com/api/admin/extension/blockingTasks/
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.blockingTaskList+xml
...
<vmext:BlockingTaskReferences
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   ... >
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.vmwExtension+xml"
      href="https://vcloud.example.com/api/admin/extension" />
   <vcloud:Reference
      type="application/vnd.vmware.admin.blockingTask+xml"
      name="vdcUpdateTemplate"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/28" />
   <vcloud:Reference
      type="application/vnd.vmware.admin.blockingTask+xml"
      name="vdcComposeVapp"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25" />
   <vcloud:Reference
      type="application/vnd.vmware.admin.blockingTask+xml"
      name="vdcUploadMedia"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/12" />
</vmext:BlockingTaskReferences>
```

Take Action on a Blocking Task

The `BlockingTask` element includes links that you can use to take action on a blocking task.

A `BlockingTask` element is primarily a collection of `Link` elements that allow you to take action on the task. When a user requests an operation that is configured to create a blocking task, the system sends a message about the task to the configured AMQP broker, and also creates a reference to the task in the cloud's `BlockingTaskReferences` container. A system administrator can retrieve the list of `BlockingTask` elements by making a GET request to the system's `extension/blockingTasks` link. See Monitor Blocking Tasks.

After authenticating to the cloud as a system administrator, the AMQP client can retrieve a blocking task. The AMQP client makes a GET request to a URL that the task creates by appending the value of the `id` attribute of the task to the `entityResolver` URL in the `Notification`. See Notification Message Format.

The following actions are allowed:

resume

Unblock the task and allow it to continue.

abort

End the task, cleaning up any transient objects that it created. Task status is set to ABORTED.

fail

End the task, setting the status of any transient objects that it created to ERROR. Task status is set to ERROR.

updateProgress

Reset the timeout value and timeout action for an active task. Use this action to keep the task alive when it might become subject to a timeout action.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the list of active blocking tasks.

   See Monitor Blocking Tasks. If you are using an AMQP client to handle blocking tasks, skip this step. Each blocking task creates its own AMQP message, which contains a reference to the `BlockingTask`.

2 Retrieve an individual `BlockingTask`.

   See the request portion of Handling a Blocking Task.

3 Make a request.

Action	Request
resume	POST a `BlockingTaskOperationParams` element to the `Link` where `rel="resume"`
abort	POST a `BlockingTaskOperationParams` element to the `Link` where `rel="abort"`
fail	POST a `BlockingTaskOperationParams` element to the `Link` where `rel="fail"`
updateProgress	POST a `BlockingTaskUpdateProgressParams` element to the `Link` where `rel="updateProgress"`

Example: Handling a Blocking Task

This request shows how to retrieve a blocking task without using an AMQP client. Notification Message Format shows how to retrieve the same task using information in the AMQP message.

Request:

```
GET https://vcloud.example.com/api/admin/extension/blockingTask/25
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.blockingTask+xml
...
<vmext:BlockingTask
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   status="active"
   timeoutDate="2011-05-07T16:25:18.857+03:00"
   timeoutAction="abort"
   createdTime="2011-05-02T16:25:18.857+03:00"
   name="importSingletonTemplate"
   id="urn:vcloud:blockingTask:25"
   type="application/vnd.vmware.admin.blockingTask+xml"
   href="https://vcloud.example.com/api/admin/extension/blockingTask/25">
   <vcloud:Link
      rel="resume"
      type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25/action/resume" />
   <vcloud:Link
      rel="abort"
      type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25/action/abort" />
   <vcloud:Link
      rel="fail"
      type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25/action/fail" />
   <vcloud:Link
      rel="updateProgress"
      type="application/vnd.vmware.admin.BlockingTaskUpdateProgressOperationParams+xml"
      href="https://vcloud.example.com/api/admin/extension/blockingTask/25/action/
updateProgress" />
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.vcloud.task+xml"
      href="https://vcloud.example.com/api/task/42" />
   <vcloud:Organization
      type="application/vnd.vmware.admin.organization+xml"
      name="example"
      href="https://vcloud.example.com/api/admin/org/97" />
   <vcloud:User
      type="application/vnd.vmware.admin.user+xml"
      name="system"
      href="https://vcloud.example.com/api/admin/user/80" />
   <vcloud:TaskOwner
      type="application/vnd.vmware.vcloud.vAppTemplate+xml"
      name=""
      href="https://vcloud.example.com/api/vAppTemplate/vappTemplate-89" />
</vmext:BlockingTask>
```

The following request allows the task to resume with a message indicating administrative approval.

```
POST https://vcloud.example.com/api/admin/extension/blockingTask/25/action/resume
Content-Type: application/vnd.vmware.admin.blockingTaskOperationParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<BlockingTaskOperationParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5" >
   <Message>Approved by system administrator.</Message>
</BlockingTaskOperationParams>
```

Extend The Timeout Expiration of an Active Task

You can use the `updateProgress` link in a `BlockingTask` to extend the expiration time of an active task.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the list of active blocking tasks.

  See Monitor Blocking Tasks. If you are using an AMQP client to handle task extension requests, skip this step. Each blocking task creates its own AMQP message, which contains a reference to the `BlockingTask` mentioned in BlockingTask.

2 Retrieve an individual `BlockingTask`.

  See the request portion of Handling a Blocking Task.

3. Provide a new timeout value, relative to now, for the task.

   Create a BlockingTaskUpdateProgressParams element that specifies the number of milliseconds until the task times out. See Extend The Timeout Expiration of an Active Task.

4. POST the BlockingTaskUpdateProgressParams to the updateProgress URL from the BlockingTask.

   The new timeout value is set to now (the time when the updateProgress request is executed) plusTimeoutValueInMilliseconds.

Example: Extend The Timeout Expiration of an Active Task

This request resets the expiration time of the BlockingTask shown in Handling a Blocking Task to ten minutes after the request is processed.

Request:

```
POST https://vcloud.example.com/api/admin/extension/blockingTask/34/action/updateProgress
Content-Type: application/vnd.vmware.admin.blockingTaskUpdateProgressOperationParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<BlockingTaskUpdateProgressParams
   xmlns="http://www.vmware.com/vcloud/extension/v1.5">
   <Message>Giving you ten more minutes...</Message>
   <TimeoutValueInMilliseconds>600000</TimeoutValueInMilliseconds>
</BlockingTaskUpdateProgressParams>
```

The response includes the entire BlockingTask and shows the new value of the timeoutDate attribute. The value assumes that the request was made at time 2011-05-11T11:50:55. This example omits most of the response.

Response:

```
200 OK
Content-Type: application/vnd.vmware.admin.blockingTask+xml
...
<vmext:BlockingTask
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   status="active"
   timeoutDate="2011-05-11T12:00:55.857+03:00"
   ...
</vmext:BlockingTask>
```

Notification Message Format

All messages that the VMware Cloud Director AMQP service sends contain an AMQP routing key and a body formatted as a JSON object or an XML Notification element.

The XML Notification element is defined in the VMware Cloud Director API schema. The routing key format is defined by the AMQP specification.

Routing Key Format

The routing key for a VMware Cloud Director AMQP message has the following form:

```
operationSuccess.entity.org.user.subType1.subType2...subTypeN.[taskName]
```

Routing key components include:

operationSuccess

A Boolean value denoting whether the operation that triggered the notification succeeded or failed.

entity

The object identifier of the object on which an operation, an event of type `com/vmware/vcloud/event/`, triggered the notification. For more information about object identifiers, see Objects, References, and Representations.

org

The object identifier of the organization that owns the affected object.

user

The object identifier of the user who made the request.

subType1-subType*N*

Each *subType* is a single component of the event type name. See Notification Types.

taskName

If *entity* is a task or blocking task, the task name is appended to the routing key.

The following routing key, in which the object identifiers are truncated to save space, is an example of a routing key that might have been created for a successful `com/vmware/vcloud/event/vapp/create` event:

```
true.dc6a-xxx.0b8a-xxx.832c-xxx.com.vmware.vcloud.event.vapp.create
```

Notification Headers

The VMware Cloud Director API defines these AMQP notification headers.

Table 11-1. Notification Headers

Header	Value
`notification.type`	See Notification Types.
`notification.entityType`	The type of VMware Cloud Director entity is associated with this notification. For example, `vm`.

Table 11-1. Notification Headers (continued)

Header	Value
`notification.entityUUID`	The object identifier of the object on which an operation, an event of type com/vmware/vcloud/event/, triggered the notification.
`notification.orgUUID`	The object identifier of the organization that owns the affected object.
`notification.userUUID`	The object identifier of the user who made the request.
`notification.operationSuccess`	A Boolean value denoting whether the operation that triggered the notification succeeded or failed.
`notification.taskUUID`	The object identifier of the task that triggered the notification. This header is present only when a task is a child of another task. Use it to correlate events across a set of tasks.
`notification.cellUUID`	The object identifier of the VMware Cloud Director cell executing the task that triggered the notification. Present only when the `notification.taskUUID` header is present.

Example: Notification Message Format

Each notification message is delivered as both XML and JSON.

Important XML `NotificationType` is deprecated in favor of the JSON format. VMware Cloud Director AMQP notifications in XML format may be removed in a future release.

JSON notifications are published on an exchange created by VMware Cloud Director with a name of the form *`prefix`*`.notifications20` where *prefix* is the value of the `AmqpPrefix` element in the system `AmqpSettings`. XML notifications are published on the AMQP exchange specified in the system AMQP settings.

Here is an example of a message generated by a blocking task, formatted as a JSON object.

```
{
   "eventId" : "a1440dd8-60ae-46c7-b216-44693bc00c90",
   "type" : "com/vmware/vcloud/event/blockingtask/create",
   "timestamp" : "2011-06-18T14:33:27.787+03:00",
   "operationSuccess" : true,
   "user" : "urn:vcloud:user:44",
   "userName" : "Bob",
   "org" : "urn:vcloud:org:70",
   "orgName" : "Finance",
   "entity" : "urn:vcloud:blockingTask:25",
   "entityName" : "vdcComposeVapp",
   "entityType" : "com.vmware.vcloud.entity.blockingTask",
   "taskOwner" : "urn:vcloud:vapp:26"
}
```

To get more information about the blocking task that generated the notification, a system administrator use the value of the `entity` key in a request to the `entityResolver` URL. See Retrieve an Object as an Entity.

This request retrieves the blocking task that generated this notification.

```
GET https://vcloud.example.com/api/entity/urn:vcloud:blockingtask:25
```

The response to this request is identical to the one shown in the response portion of Handling a Blocking Task.

Notification Types

The value of the `type` attribute of a VMware Cloud Director notification is a string of the form `com/vmware/vcloud/event/`*`object-type`*`/`*`event-type`*. Notification types can be grouped based on the object type affected by the event.

User, Group, Role, and Session Events

Table 11-2. User, Group, Role, and Session Events

Type (com/vmware/vcloud/event/)	Description
session/login	A login session was created.
user/import	A user was imported from LDAP.
user/remove	An imported user was removed from the organization.
user/modify	One or more properties of a user were modified.
user/lockout	An account was locked based on the organization's password policy settings.
user/unlock	A locked account was unlocked.
user/lock_expired	The lock on an account has expired.
user/create	A local user was created in an organization.
user/delete	A local user was removed from the organization.
group/import	A group was imported from LDAP.
group/remove	A group was removed from an organization.
role/create	A new role was created.
role/modify	An existing role was modified.
role/delete	A role was deleted.

Organization, Network, Catalog, and VDC Events

Table 11-3. Organization, Network, Catalog, and VDC Events

Type (com/vmware/vcloud/event/)	Description
org/create	An organization was created.
org/modify	An organization was modified.
org/delete	An organization was deleted.
network/create	A network was created.
network/modify	A network was modified.
network/delete	A network was deleted.
network/deploy	A network was deployed.
network/undeploy	A network was undeployed.
catalog/create	A catalog was created.
catalog/delete	A catalog was deleted.
catalog/modify	One or more properties of a catalog were modified
catalog/publish	A catalog was published.
catalogItem/create	An item was added to a catalog.
catalogItem/delete	An item was removed from a catalog.
vdc/create_request	A request to create a VDC was blocked pending administrative action.
vdc/create	A VDC was created.
vdc/modify	One or more properties of a VDC was modified.
vdc/delete_request	A request to delete a VDC was blocked pending administrative action.
vdc/delete	A VDC was deleted.
vdc/fast_provisioning/modify	The `UsesFastProvisioning` value of a VDC was modified.
vdc/thin_provisioning/modify	The `IsThinProvision` value of a VDC was modified.

vApp, vApp Template, Vm, and Media Events

Table 11-4. vApp, vApp Template, Vm, and Media Events

Type (com/vmware/vcloud/event/)	Description
vappTemplate/create	A vApp template was created.
vappTemplate/import	A virtual machine was imported from vSphere as a vApp template.
vappTemplate/modify	One or more properties of a vApp template were modified.
vappTemplate/delete	A vApp template was deleted.
vappTemplate/create_request	A request to create a vApp template was blocked pending administrative action.
vappTemplate/import_request	A request to import a vApp template was blocked pending administrative action.
vappTemplate/modify_request	A request to modify a vApp template was blocked pending administrative action.
vappTemplate/delete_request	A request to delete a vApp template was blocked pending administrative action.
vapp/create	A vApp was created (instantiated)
vapp/import	A virtual machine was imported from vSphere as a vApp.
vapp/modify	One or more properties of a vApp were modified.
vapp/delete	A vApp was deleted.
vapp/deploy	A vApp was deployed.
vapp/undeploy	A vApp was undeployed.
vapp/runtime_lease_expiry	The runtime lease of a vApp has expired.
vapp/create_request	A request to instantiate a vApp template was blocked pending administrative action.
vapp/import_request	A request to import a vApp was blocked pending administrative action.
vapp/modify_request	A request to modify a vApp was blocked pending administrative action.
vapp/delete_request	A request to delete a vApp was blocked pending administrative action.
vapp/deploy_request	A request to deploy a vApp was blocked pending administrative action.
vapp/undeploy_request	A request to undeploy a vApp was blocked pending administrative action.
vm/create_request	A request to create a virtual machine was blocked pending administrative action.

Table 11-4. vApp, vApp Template, Vm, and Media Events (continued)

Type (com/vmware/vcloud/event/)	Description
vapp/quarantine_reject	An uploaded OVF was rejected after quarantine.
vapp/upload_timeout	An OVF upload has timed out.
vapp/lease_expiration_changed	The lease expiration of a vApp has changed.
vm/ip_address_changed	The IP address of a virtual machine has changed.
vm/create	A virtual machine was created by instantiating a vApp.
vm/modify_request	A request to modify a virtual machine was blocked pending administrative action.
vm/modify	One or more properties of a virtual machine were modified.
vm/delete	A virtual machine was deleted.
vm/change_state	The power state of a virtual machine has changed.
vm/deploy_request	A request to deploy a virtual machine was blocked pending administrative action.
vm/deploy	A virtual machine was deployed.
vm/undeploy_request	A request to undeploy a virtual machine was blocked pending administrative action.
vm/undeploy	A virtual machine was undeployed.
vm/consolidate_request	A request to consolidate a virtual machine was blocked pending administrative action.
vm/consolidate	A virtual machine was consolidated.
vm/relocate_request	A request to relocate a virtual machine was blocked pending administrative action.
vm/relocate	A virtual machine was relocated.
media/create	A media object was created by upload or import.
media/import	A media object was imported.
media/modify	One or more properties of a media object were modified.
media/delete	A media object was deleted.
media/create_request	A request to create a media object was blocked pending administrative action.
media/import_request	A request to import a media object was blocked pending administrative action.
media/modify_request	A request to modify a media object was blocked pending administrative action.

Table 11-4. vApp, vApp Template, Vm, and Media Events (continued)

Type (com/vmware/vcloud/event/)	Description
media/delete_request	A request to delete a media object was blocked pending administrative action.
media/upload_timeout	A media upload has timed out.
media/quarantine_reject	An uploaded media object was rejected after quarantine.

Other System Events

Table 11-5. Other System Events

Type (com/vmware/vcloud/event/)	Description
providerVdc/create_request	A request to create a provider VDC was blocked pending administrative action.
providerVdc/create	A provider VDC was created.
providerVdc/modify	One or more properties of a provider VDC were modified.
providerVdc/delete_request	A request to delete a provider VDC was blocked pending administrative action.
providerVdc/delete	A provider VDC was deleted.
vc/create	A vCenter server was registered.
vc/modify	One or more properties of a registered vCenter server were modified.
vc/delete	A registered vCenter server was registered.
task/create	A task was created.
task/start	A non-blocking task has started or a blocking task has resumed.
task/abort	A task was aborted.
task/complete	A task has completed.
task/fail	A task has failed.
task/update	Task progress was updated.
blockingtask/create	A task was blocked and a notification created.
blockingtask/resume	A blocking task was resumed.
blockingtask/abort	A blocking task was aborted.
blockingtask/fail	A blocking task was failed.

Table 11-5. Other System Events (continued)

Type (com/vmware/vcloud/event/)	Description
datastore/modify	One or more properties of a datastore object were modified.
datastore/delete	A datastore object was deleted.

Extending VMware Cloud Director 12

VMware Cloud Director provides two extension mechanisms. Extension services can populate VMware Cloud Director objects with service-specific links. Object extensions can register interest in specific VMware Cloud Director operation and become part of the processing logic employed by those operations.

VMware Cloud Director object extensions are similar to VMware Cloud Director extension services in that both make use of the same AMQP-based message-queuing mechanisms. But where extension services typically operate by inserting service-specific links in VMware Cloud Director objects, object extensions operate by registering interest in one or more phases of a VMware Cloud Director procedure such as vApp placement or instantiation and, using information from the system, influencing the outcome of those processes in ways that reflect site-specific or customer-specific requirements.

Read the following topics next:

VMware Cloud Director Object Extensions

VMware Cloud Director object extensions are external applications that can participate in, influence, or override the logic that VMware Cloud Director applies to workflows like vApp instantiation and placement.

An object extension can participate as a peer of the system in the logic that determines the outcome of these workflows. For example, an object extension might use information provided by the system to place a VM on a specific host or assign it a specific storage profile. Operation of object extensions is transparent to system users. An extension can also allow the VMware Cloud Director procedures in which it participates to run to completion unmodified.

Object Extensibility Framework

VMware Cloud Director object extensions are implemented in a framework that has three components:

- The VMware Cloud Director API supports VMware Cloud Director object extensions with two kinds of operations:
  - Object registration requests, which register an object extension with the system.
  - Selector extension requests, which select system objects to extend and specify the extension points and execution phases in which the extension participates.
- Communication between the system and an object extension is handled using the system AMQP service described in #unique_231. Extensions must create their own exchanges.
- Extension points themselves are defined in the system. When an object extension registers with VMware Cloud Director, it can specify one more of these extension points, expressed as a URN:
  - `urn:selector:providerVdc`
  - `urn:selector:organizationVdc`
  - `urn:selector:organization`

An object extension that operates on a subset of these extension points can specify that subset at registration to prevent it from being associated with extension points outside the subset. An object extension that operates on all extension points does not need to specify any of them at registration.

All other operations are implemented by the extensions themselves, which read messages from their AMQP exchange and provide responses, as AMQP messages, to VMware Cloud Director.

For additional information and code examples, see VMware Sample Exchange at https://developercenter.vmware.com/samples/.

Extensibility Phases

Extensible operations are distinguished by having life cycle events such as create, update, and delete mapped to an extensibility phas e. An object extension can participate in zero or more phases of an extensible VMware Cloud Director operation. Each phase is identified by a URN.

The system collects the internal information that it uses when executing a phase and sends it to the extension as an AMQP message. If specified by the extension, the phase waits for a reply before proceeding (see Message Handling). By sending a reply that modifies values that the phase supplies, the extension can affect the outcome of the phase. XML schema definition files for the messages sent by each phase are available on the VMware Sample Exchange at https://developercenter.vmware.com/samples/.

Each phase has an inherent cardinality that establishes the number of extensions that can participate in phase operations. Some operations such as VM placement, which must take into account factors such as host configuration, affinities, and storage capabilities, can accommodate participation by multiple extensions and have a cardinality greater than 1. Others, such as creation of a VM, are not candidates for participation from multiple extensions, and have a cardinality of exactly 1.

Table 12-1. Extensibility Phases

Name	Cardinality	Timeout in Seconds	Summary
`urn:extensionPoint:vm:gatherRequirements`	1-*n*	5	Gathers virtual machine placement requirements, including CPU and memory configurations, storage IOPS, and datastore affinity and provides them to the extension
`urn:extensionPoint:vm:calculateSolution`	1-*n*	60	Calculates a placement solution and provides it to the extension
`urn:extensionPoint:vm:create`	1	60	Creates a virtual machine

Message Handling

An object extension can specify how it wants to handle messages sent by a phase. Three types of message handling are supported.

async

async messages provide a stream of information as the system proceeds with phase execution. Extensions that specify async message handling are informed of phase operations but cannot influence the outcome of a phase.

blocking

blocking messages force the execution of the phase to block until the extension takes some action. Extensions that specify blocking message handling must reply to each message within the specified timeout interval or the system proceeds with the operation.

needsApproval

needsApproval messages force the execution of the phase to block until approved by a system administrator.

Selector Associations and Extension Policies

When it creates a SelectorExtension, an object extension can express interest in all objects of a certain type, or in a specific object. An object extension that selects an object type affects all instances of that type. Object extensions that select an object instance affect only that instance, and override the operations of any extension that might be in place for that instance's type.

Selector extension request URLs have one of the following forms, where *type* is one of:

- `providervdcs`
- `vdcs`
- `orgs`

In this table:

- *API-URL* is a URL of the form `https://vcloud.example.com/api`.
- *id* is a unique identifier in the form of a UUID, as defined by RFC 4122.

Table 12-2. Selector Extension Request URLs

HTTP Verb	Request URL Prototype	Summary
GET	*API-URL*/admin/*type*/extension	Retrieve all extensions that are associated with the default selector policy for *type*.
POST	*API-URL*/admin/*type*/extension	Add a selector extension to the default policy for *type*.
GET	*API-URL*/admin/*type*/extension/*extension-ID*	Retrieve extension *extension-ID* associated with the default selector policy for *type*.
PUT	*API-URL*/admin/*type*/extension/*extension-ID*	Update extension *extension-ID* associated with the default selector policy for *type*.
DELETE	*API-URL*/admin/*type*/extension/*extension-ID*	Delete extension *extension-ID* from the default selector policy for *type*.
GET	*API-URL*/admin/*type*/*selector-ID*/extension/	Retrieve all extensions associated with the specified *selector-ID* and *type*.
POST	*API-URL*/admin/*type*/*selector-ID*/extension/	Add a selector extension to the specified *selector-ID* to update the default selector policy for *type*.

Extension Workflow

A typical extension implements a workflow similar to this one:

1. Use the VMware Cloud Director API to create an extension-specific exchange on the system AMQP service. See #unique_231.
2. Set up a listener for that exchange.
3. Use the VMware Cloud Director API to register with the system. See Register an Object Extension.
4. Use the VMware Cloud Director API to create one or more selector extensions for specific objects or types of objects. See Create a Selector Extension.
5. Listen on the AMQP exchange for messages from extension points, and take appropriate actions. For example:
   - Reply to the extension point with a message that modifies some of the values included in a previous message

- Use the VMware Cloud Director API to create or update objects based on the results of message analysis or processing

An extension continues handling the stream of messages from extension points if the system or the extension is active.

Related Videos

(Object Extensibility in VMware Cloud Director)

Register an Object Extension

Register an object extension to specify its namespace, vendor, and AMQP exchange name.

An object extension typically authenticates with the VMware Cloud Director API as a system administrator, then registers with VMware Cloud Director by POSTing an `ObjectExtension` element to the system's `.../api/admin/extension/object` URL. An `ObjectExtension` element must include the following elements.

`Namespace`

The object extension namespace.

`Vendor`

The object extension vendor.

Important The combination of `Namespace` and `Vendor` must be unique among all extension services and object extensions registered in the system. If a service tries to register an object extension that specifies a `Namespace` and `Vendor` that are already registered with this VMware Cloud Director installation, registration fails. Vendor and namespace names that follow the naming convention used for Java packages are more likely to be unique.

`Selectors`

Zero or more `Selector` elements that specify the extension points in which the extension has an interest. If you register an object extension with no `Selectors`, the system assumes that the extension is interested in all extension points.

`Exchange`

The AMQP exchange name that VMware Cloud Director must use when routing messages to the extension. The extension must create the specified exchange on the AMQP service that VMware Cloud Director uses. The exchange type must be `direct`. Exchange names must be globally unique.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the cloud.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/admin/extension
   ```

2 Examine the response to find the `Link` for adding object extensions.

   This `Link` is present in the `VMWExtension` element, and has the following form.

   ```
   <vcloud:Link
      rel="add"
      href="https://vcloud.example.com/api/admin/extension/object"
      type="application/vnd.vmware.admin.extensibility.object+xml" />
   ```

3 Construct an `ObjectExtension` element.

   See the request portion of Register an Object Extension.

4 POST the `ObjectExtension` element to the URL described in Step 2.

   See the request portion of Register an Object Extension.

Example: Register an Object Extension

This example registers an object extension for the `providerVdc` extension point but does not enable it. The example assumes that an AMQP exchange named `example-exchange` has been created.

Request:

```
POST https://vcloud.example.com/api/admin/extension/object
Content Type: application/vnd.vmware.admin.extensibility.object+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ObjectExtension
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud_v1.5="http://www.vmware.com/vcloud/v1.5">
   <Namespace>vcdx.vmdeploy</Namespace>
   <Enabled>false</Enabled>
   <Exchange>example-exchange</Exchange>
   <Vendor>com.example</Vendor>
   <Selectors>
      <Selector name="urn:selector:providerVdc" />
   </Selectors>
</ObjectExtension>
```

The system registers the extension and returns an `ObjectExtension` element that includes information derived from the contents you POSTed, and a set of `Link` elements that you can use to access or modify the new object extension. The AMQP message `ContentType`, which was not specified in the request, is returned with the default value, `XML`.

Response:

```
201 Created
Content Type: application/vnd.vmware.admin.service+xml
...

<vmext:ObjectExtension
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   id="83...0d39"
   href="https://vcloud.example.com/api/admin/extension/object/83...0d39" >
   <vcloud:Link
      rel="up"
      href="https://vcloud.example.com/api/admin/extension/object"
      type="application/vnd.vmware.admin.extensibility.objects+xml" />
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/object/83...0d39"
      type="application/vnd.vmware.admin.extensibility.object+xml" />
    <vcloud:Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/extension/object/83...0d39"
      type="application/vnd.vmware.admin.extensibility.object+xml" />
   <vcloud:Link
      rel="down"
      href="https://vcloud.example.com/api/admin/extension/object/83...0d39/
selectorExtensions"
      type="application/vnd.vmware.admin.extensibility.selectors+xml" />
   <vmext:Namespace>example-object</vmext:Namespace>
    <vmext:Enabled>false</vmext:Enabled>
    <vmext:Exchange>example-exchange</vmext:Exchange>
   <vmext:ContentType>XML</vmext:ContentType>
   <vmext:Vendor>com.example</vmext:Vendor>
   <vmext:Selectors>
      <vmext:Selector name="urn:selector:providerVdc" />
   </vmext:Selectors>
</vmext:ObjectExtension>
```

Retrieve or Update an Object Extension

You can update an object extension registration by making a PUT request to its `edit` URL and supplying a modified an `ObjectExtension` in the request body.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the object extension.

Use a request like this one, where *ID* is the ID of the extension that you want to retrieve.

```
GET https://vcloud.example.com/api/admin/extension/object/ID
```

If you do not know the ID of the object extension, you can retrieve a list of all registered object extensions with a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/object
```

2. Examine the response to find the object extension's `edit` link.

```
<vcloud:Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/extension/object/83...0d39"
      type="application/vnd.vmware.admin.extensibility.object+xml" />
```

3. Update the retrieved `ObjectExtension`.

4. Make a PUT request to the URL in the edit link, supplying the modified `ObjectExtension` in the request body. See Update an Object Extension

Example: Update an Object Extension

This example updates the object extension created in Register an Object Extension to enable it by setting the value of `Enabled` to `true`.

Request:

```
PUT https://vcloud.example.com/api/admin/extension/object
Content Type: application/vnd.vmware.admin.extensibility.object+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<ObjectExtension
   xmlns="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud_v1.5="http://www.vmware.com/vcloud/v1.5">
   <Namespace>vcdx.vmdeploy</Namespace>
   <Enabled>true</Enabled>
   <Exchange>example-exchange</Exchange>
   <Vendor>com.example</Vendor>
   <Selectors>
      <Selector name="urn:selector:providerVdc" />
   </Selectors>
</ObjectExtension>
```

The system updates the extension and returns an updated `ObjectExtension` element similar to the one shown in Register an Object Extension.

Response:

```
200 OK
Content Type: application/vnd.vmware.admin.service+xml
...
<vmext:ObjectExtension
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   id="83...0d39"
   href="https://vcloud.example.com/api/admin/extension/object/83...0d39" >
   ...
```

```
    <vmext:Namespace>example-object</vmext:Namespace>
    <vmext:Enabled>true</vmext:Enabled>
    <vmext:Exchange>example-exchange</vmext:Exchange>
    <vmext:ContentType>XML</vmext:ContentType>
    <vmext:Vendor>com.example</vmext:Vendor>
    ...
</vmext:ObjectExtension>
```

Create a Selector Extension

A selector extension associates a registered object extension with one or more of the phases of an extensible object type.

Associating a selector extension with an object type or instance and registered object extension enables the object extension to receive messages when an operation on the type or instance reaches an extensible phase of its execution. A selector extension can be associated with one or more object extensions.

Prerequisites

This operation is restricted to system administrators.

You must provide the ID of a registered object extension as part of the request to create a selector extension. That ID is returned when you create the object extension. If you don't know the ID of the object extension, you can retrieve a list of all registered object extensions with a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/object
```

Procedure

1 Construct an `ObjectExtension` element.

  The `ObjectExtensionId` element in the request must specify the ID of the object extension for which you are creating this selector extension. See the request portion of Create a Selector Extension for a Registered Object Extension.

2 POST the `ObjectExtension` element to the URL for the object that you wish to extend.

  The URL can refer to a specific object or all objects of a certain type. See the request portion of Create a Selector Extension for a Registered Object Extension.

Example: Create a Selector Extension for a Registered Object Extension

This example adds a selector extension to the object extension registered in Register an Object Extension. This request is appropriate for an extension that must gather and analyze Provider VDC requirements associated with the creation of a virtual machine, but not influence the operation of the phase. The selector specifies the `urn:extensionPoint:vm:gatherRequirements` phase and `async` execution, so the system does not block awaiting a response. Because the phase has an `optional` attribute with a value of `true` phase execution continues even if the extension fails.

If the extension is to influence the outcome of the phase by replying with modified requirements, it must set `type` to `blocking` and be prepared to reply within the specified time-out interval.

In this example, the selector extension sets `Priority` to `1`. The system uses the `Priority` value to control the order of execution when multiple extensions have registered interest in a phase. Note that a `Priority` of `1` does not always mean that the extension is called first. For some extensions, such as those that select the `urn:extensionPoint:vm:calculateSolution` phase and affect placement, the highest priority position in the execution order is the final one. This approach ensures that any placement solution offered by the extension cannot be superseded by a solution offered by a different extension.

A request like this one sets this `SelectorExtension` as the default extension policy for all Provider VDCs in the system.

Request:

```
POST https://vcloud.example.com/api/admin/providervdcs/extension
Content Type: application/vnd.vmware.admin.extensibility.selector+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<SelectorExtension
   xmlns="http://www.vmware.com/vcloud/v1.5" >
   <Enabled>false</Enabled>
   <Priority>1</Priority>
   <ObjectExtensionId>83...0d39</ObjectExtensionId>
   <Phases>
      <Phase
         name="urn:extensionPoint:vm:gatherRequirements"
         type="async"
         optional="true" />
   </Phases>
</SelectorExtension>
```

To apply this policy to a single Provider VDC, use a request of the following form:

```
POST https://vcloud.example.com/api/admin/providervdc/ID/extension
```

The system creates the selector extension and returns a `SelectorExtension` element that includes information derived from the contents you POSTed, and a set of `Link` elements that you can use to access or modify the new object extension.

Response:

```
201 Created
Content Type: application/vnd.vmware.admin.extensibility.selector+xml
...
<SelectorExtension
   xmlns="http://www.vmware.com/vcloud/v1.5"
   id="b3...f8ee"
   href="https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee"
   ... >
   <Link
```

```
      rel="remove"
      href="https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee" />
   <Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee"
      type="application/vnd.vmware.admin.extensibility.selector+xml" />
   <Link
      rel="up"
      href="https://vcloud.example.com/api/admin/providervdcs/extension"
      type="application/vnd.vmware.admin.extensibility.selectors+xml" />
   <Enabled>false</Enabled>
   <Priority>1</Priority>
   <ObjectExtensionId>83...0d39</ObjectExtensionId>
   <Phases>
      <Phase
         name="urn:extensionPoint:vm:gatherRequirements"
         optional="true"
         type="ASYNC" />
    </Phases>
</SelectorExtension>
```

The selector extension is added to the set of selector extensions associated with the specified object extension. You can retrieve a list of those selector extensions with a request to the object extension's `extensibility.selectors` link.

```
GET https://vcloud.example.com/api/admin/extension/object/83...0d39/selectorExtensions
...
<SelectorExtension
   xmlns="http://www.vmware.com/vcloud/v1.5"
   id="b3...f8ee"
   href="https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee" ...>
   ...
</SelectorExtension>
```

Retrieve or Update a Selector Extension

You can update a selector extension by making a PUT request to its `edit` URL and supplying a modified `SelectorExtension` in the request body.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the selector extension.

   Use a request like this one, where *ID* is the ID of the extension that you want to retrieve.

   ```
   GET https://vcloud.example.com/api/admin/type/extension/ID
   ```

2 Examine the response to find the object extension's `edit` link.

```
<vcloud:Link
      rel="edit"
      href="https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee"
      type="application/vnd.vmware.admin.extensibility.selector+xml" />
```

3 Update the retrieved `SelectorExtension`.

4 Make a PUT request to the URL in the edit link, supplying the modified `SelectorExtension` in the request body. See Update a Selector Extension

Example: Update a Selector Extension

This example updates the selector extension created in Create a Selector Extension for a Registered Object Extension to enable it by setting the value of `Enabled` to `true`.

Request:

```
PUT https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee
Content Type: application/vnd.vmware.admin.extensibility.selector+xml
...
Content Type: application/vnd.vmware.admin.extensibility.selector+xml
<?xml version="1.0" encoding="UTF-8"?>
<SelectorExtension
   xmlns="http://www.vmware.com/vcloud/v1.5" >
   <Enabled>true</Enabled>
   <Priority>1</Priority>
   <ObjectExtensionId>83...0d39</ObjectExtensionId>
   <Phases>
      <Phase
         name="urn:extensionPoint:vm:gatherRequirements"
         type="async"
         optional="true" />
   </Phases>
</SelectorExtension>
```

The system updates the extension and returns an updated `SelectorExtension` element similar to the one shown in Create a Selector Extension for a Registered Object Extension.

Response:

```
200 OK
Content Type: application/vnd.vmware.admin.extensibility.selector+xml
...
<SelectorExtension
   xmlns="http://www.vmware.com/vcloud/v1.5"
   id="b3...f8ee"
   href="https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee"
   ... >
   <Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee" />
   <Link
```

```
      rel="edit"
      href="https://vcloud.example.com/api/admin/providervdcs/extension/b3...f8ee"
      type="application/vnd.vmware.admin.extensibility.selector+xml" />
   <Link
      rel="up"
      href="https://vcloud.example.com/api/admin/providervdcs/extension"
      type="application/vnd.vmware.admin.extensibility.selectors+xml" />
   <Enabled>true</Enabled>
   <Priority>1</Priority>
   <ObjectExtensionId>83...0d39</ObjectExtensionId>
   <Phases>
      <Phase
         name="urn:extensionPoint:vm:gatherRequirements"
         optional="true"
         type="ASYNC" />
    </Phases>
</SelectorExtension>
```

VMware Cloud Director Extension Services

VMware Cloud Director and the VMware Cloud Director API include a framework for integration of extension services that VMware Cloud Director API client can access as though they were native services. In addition to service-specific objects or operations they provide, extension services can implement new operations for existing API objects.

A VMware Cloud Director extension service is a program that presents a REST interface to VMware Cloud Director API clients. When you register an extension service with theVMware Cloud Director API, you specify one or more URL patterns that the VMware Cloud Director REST service treats as extension requests. When it receives an extension request, the VMware Cloud Director REST service creates an AMQP notification with a service-specific exchange and routing key, and sends it to the VMware Cloud Director AMQP service. Each extension service subscribes to AMQP notifications that have its service-specific routing key. A service processes its notifications, takes whatever actions they require, and returns a response to the AMQP service, where the VMware Cloud Director REST service retrieves it and uses its contents to generate a response to the client that made the request.

Message Routing

Extension services use the VMware Cloud Director AMQP service to communicate with VMware Cloud Director. Every extension service must register a unique AMQP routing key, which VMware Cloud Director prepends to AMQP messages destined for that service. To collect replies from services, VMware Cloud Director creates a single reply exchange for all services, creates a separate reply queue for each cell, and binds each of those queues to the reply exchange.

VMware Cloud Director extension services can also be VMware Cloud Director API clients, authenticating to the VMware Cloud Director API and making their own REST requests to the VMware Cloud Director API URL. This type of interaction is required when creating tasks and events that track the progress of requests made to the service. It is also required by services that operate on VMware Cloud Director objects like vApps and virtual machines.

Creating Events and Tasks

The VMware Cloud Director API extension framework implements operations that allow an extension service to create and update an organization's lists of tasks and events, so the status of asynchronous events running in extension services can be displayed with the same kinds of information posted by native services.

Authorization Framework

All requests to extension services are processed through the VMware Cloud Director authentication framework. A user making a request to an extension service must be authenticated by VMware Cloud Director as a system administrator or a member of a VMware Cloud Director organization.

An extension service can add service-specific rights and associate those rights with operations on its own objects or with operations it adds to VMware Cloud Director API objects .

Service APIs

An extension service can define its own request and response body elements if it needs to. API schema files can be specified as part of service registration or can be added later. Schema files can reside at any location that is reachable by the extension service.

Support for Idempotent Operations

Most requests to extension services can include an `operationKey` attribute, which is a string meant to uniquely identify the operation so that if it is invoked multiple times, the result will be the same as if it had been invoked only once. The following types support use of the `operationKey` attribute:

- `AclRuleType`
- `ApiDefinitionType`
- `ApiFilterType`
- `FileDescriptorType`
- `ResourceClassActionType`
- `ResourceClassType`
- `ServiceLinkType`
- `ServiceResourceType`
- `ServiceType`

Register an Extension Service

Register an extension service to specify its namespace, AMQP exchange and routing key, and URL patterns. You can specify additional service properties during registration or update them later.

An extension service typically authenticates with the VMware Cloud Director API as a system administrator, then registers itself with VMware Cloud Director by POSTing a `Service` element to the system's `.../api/admin/extension/service` URL. A `Service` element must include the following elements.

`Namespace`

The service namespace, which must be unique among all registered extension services. Service namespace names that follow the naming convention used for Java packages (for example, `com.example.service.backup`) are more likely to be unique. If a service tries to register a namespace that is already registered with this VMware Cloud Director installation, registration fails.

`RoutingKey`

The AMQP routing key that VMware Cloud Director must use when routing messages to the service.

`Exchange`

The AMQP exchange name that VMware Cloud Director must use when routing messages to the service. The service must create the specified exchange on the AMQP service that VMware Cloud Director uses. The exchange type must be `direct`.

`ApiFilters`

Specifies one or more URL patterns that the VMware Cloud Director REST service must treat as extension requests. URL patterns can include regular expressions that `java.util.regex.Pattern` recognizes. See Create an API Filter for an Extension Service.

Registration can also specify the following optional properties:

- Definitions of `Link` elements that the service adds to the representations of VMware Cloud Director API objects. See Service-Specific Links.
- Authorization framework for controlling access to the service's objects and operations. See Authorization Framework for Extension Service Operations.
- Locations of schema files if the service provides its own API. See REST APIs for Extension Services

You can also create or update these properties after you register the service.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the cloud.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension
```

2 Examine the response to find the `Link` for adding extension services.

This URL is present in the `VMWExtension` element, and has the following form.

```
<vcloud:Link
   rel="down"
   type="application/vnd.vmware.admin.extensionServices+xml"
   href="https://vcloud.example.com/api/admin/extension/service"/>
```

3 Construct a `Service` element.

See the request portion of Register an Extension Service.

4 POST the `Service` element to the URL described in Step 2.

See the request portion of Register an Extension Service.

Example: Register an Extension Service

This request registers an extension service named `SDK-BackupExtension`. The request specifies the service namespace and routing key, and several URL patterns to be used as API filters. Messages for the service are sent to the AMQP exchange named `sdkext` with routing key `sdkbackup`.

Note If the specified exchange does not exist on the AMQP service that VMware Cloud Director uses, an internal server error occurs whenever VMware Cloud Director receives a request that matches one of the service's API filters.

This request also includes several `ServiceLink` elements. For information about the contents of these elements, see Service-Specific Links.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service
Content Type: application/vnd.vmware.admin.service+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:Service
   xmlns="http://www.vmware.com/vcloud/v1.5"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   name="SDK-BackupExtension">
   <vmext:Namespace>org.example.vcd.backup</vmext:Namespace>
   <vmext:Enabled>true</vmext:Enabled>
   <vmext:AuthorizationEnabled>true</vmext:AuthorizationEnabled>
   <vmext:RoutingKey>backup</vmext:RoutingKey>
   <vmext:Priority>50</vmext:Priority>
   <vmext:Exchange>sdkext</vmext:Exchange>
   <vmext:ApiFilters>
      <vmext:ApiFilter>
         <vmext:UrlPattern>(/api/org/.*/backups)|(/api/vApp/vapp-.*/backups)|(/api/vApp/
vapp-.*/action/backup)|(/api/backup/.*)</vmext:UrlPattern>
      </vmext:ApiFilter>
   </vmext:ApiFilters>
   <vmext:ServiceLinks>
      <vmext:ServiceLink>
         <vmext:LinkHref>{baseUri}org/{resourceId}</vmext:LinkHref>
         <vmext:MimeType>application/vnd.example.vcd-backup.backupList+xml</vmext:MimeType>
         <vmext:Rel>down</vmext:Rel>
         <vmext:ResourceType>application/vnd.example.vcd-backup.org+xml</vmext:ResourceType>
      </vmext:ServiceLink>
      <vmext:ServiceLink>
         <vmext:LinkHref>{baseUri}api/vApp/vapp-{resourceId}/backups</vmext:LinkHref>
         <vmext:MimeType>application/vnd.example.vcd-backup.backupList+xml</vmext:MimeType>
         <vmext:Rel>down</vmext:Rel>
         <vmext:ResourceType>application/vnd.example.vcd-backup.vApp+xml</vmext:ResourceType>
      </vmext:ServiceLink>
      <vmext:ServiceLink>
         <vmext:LinkHref>{baseUri}vApp/vapp-{resourceId}/action/backup</vmext:LinkHref>
         <vmext:MimeType>application/vnd.example.vcd-backup.createBackupParams+xml</
vmext:MimeType>
         <vmext:Rel>backup</vmext:Rel>
         <vmext:ResourceType>application/vnd.example.vcd-backup.vApp+xml</vmext:ResourceType>
      </vmext:ServiceLink>
   </vmext:ServiceLinks>
</vmext:Service>
```

The server registers the service and returns a `Service` element that includes information derived from the contents you POSTed, and a set of `Link` elements that you can use to access, remove, disable, or modify the extension service.

Response:

```
201 Created
Content Type: application/vnd.vmware.admin.service+xml
...
<vmext:Service
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="SDK-BackupExtension"
   id="urn:vcloud:externalService:45"
   type="application/vnd.vmware.admin.service+xml"
   href="https://vcloud.example.com/api/admin/extension/service/45"
   ... >
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/service/45" />
   <vcloud:Link
      rel="edit"
      type="application/vnd.vmware.admin.service+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45" />
   <vcloud:Link
```

```
      rel="rights"
      type="application/vnd.vmware.admin.rights+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/rights" />
   <vcloud:Link
      rel="down:serviceLinks"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/links" />
   <vcloud:Link
      rel="bundle:upload"
      type="application/vnd.vmware.admin.bundleUploadParams+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/localizationbundles" />
   <vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.serviceLink+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/links" />
   <vcloud:Link
      rel="down:apiFilters"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/apifilters" />
   <vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.apiFilter+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/apifilters" />
   <vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.apiDefinition+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/apidefinitions" />
   <vcloud:Link
      rel="down:apiDefinitions"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/apidefinitions" />
   <vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.resourceClass+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/resourceclasses" />
   <vcloud:Link
      rel="down:resourceClasses"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/resourceclasses" />
   <vcloud:Link
      rel="authorization:check"
      type="application/vnd.vmware.admin.authorizationCheckParams+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/authorizationcheck" />
   ...
</vmext:Service>
```

The following elements are never returned as part of a response body. Instead, they are returned as `Link` elements in the body of their container.

- `AclRules`
- `ApiDefinitions` and `Files`
- `ApiFilters`

- ResourceClassActions
- ResourceClasses
- ServiceLinks
- ServiceResources

For example, to retrieve the contents of the ServiceLinks element POSTed with the request body in this example, GET the URL in this Link:

```
<vcloud:Link
      rel="down:serviceLinks"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/links" />
```

Service-Specific Links

A service can add its own `Link` elements to the representations of VMware Cloud Director API objects. You can create these service links when you register a service. You can also add or remove the links after you register the service.

You can create multiple service-specific links as part of registering a service. After you register a service, you can add or remove individual links. Service links typically appear in the representations of all objects of a specific type, but you can constrain them to appear in a particular object of that type. Service links are not included in object representations unless the service that created them is enabled.

Note You cannot update a service link, but you can remove existing links and create new ones.

Add a Service Link

You can add a service link to an existing service.

A `ServiceLink` element must contain the following child elements:

LinkHref

The value of `href` attribute of the `Link`. This can be any URI, and can include the variables `{baseUri}` and `{resourceId}`. When constructing the `href` value of the `Link`, VMware Cloud Director replaces `{baseUri}` with the VMware Cloud Director REST API base URL, and replaces `{resourceId}` with the UUID portion of the `id` attribute value of the resource in which the `Link` is inserted. The following example might expand to the string `https://vcloud.example.com/org/17`.

```
<LinkHref>
   {baseUri}/org/{resourceId}
</LinkHref>
```

MimeType

The value, specified as a MIME content type, of the `type` attribute of the `Link`.

Rel

Defines the relationship of the link to the object that contains it. A relationship can be the name of an operation on the object, a reference to a contained or containing object, or a reference to an alternate representation of the object. The relationship value implies the HTTP verb to use when you use the link's href value as a request URL.

ResourceType

The object type, specified as a MIME content type, of the object in which the `Link` appears.

Note You can constrain the `Link` to appear in a specific resource by including a `ResourceId` element in the `ServiceLink`. This element contains the `id` of the resource in which the `Link` will appear. This resource must be of the type specified in the `ResourceType` element of the `ServiceLink`.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the service.

This request retrieves the XML representation of the service created in Register an Extension Service:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

2 Examine the response to find the `Link` for adding service links.

This `Link` has the following form:

```
<vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.serviceLink+xml"
      href="https://vcloud.example.com/api/admin/extension/service/44/links" />
```

3 Create a `ServiceLink` element that specifies the properties of the new link.

4 POST the `ServiceLink` element to the URL described in Step 2

See Add a Service Link.

Example: Add a Service Link

This request adds a `ServiceLink` to the service created in Register an Extension Service

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/links
Content-type: application/vnd.vmware.admin.serviceLink+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:ServiceLink
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   type="application/vnd.vmware.admin.serviceLink+xml">
   <vmext:LinkHref>{baseUri}vApp/vapp-{resourceId}/action/deleteBackup</vmext:LinkHref>
   <vmext:MimeType>application/vnd.example.vcd-backup.createBackupParams+xml</vmext:MimeType>
   <vmext:Rel>deleteBackup</vmext:Rel>
   <vmext:ResourceType>application/vnd.example.vcd-backup.vApp+xml</vmext:ResourceType>
</vmext:ServiceLink>
```

Response:

```
200 OK
Content-type: application/vnd.vmware.admin.serviceLink+xml
...
<vmext:ServiceLink ...>
   ...
</vmext:ServiceLink>
```

Delete a Service Link

Delete a service link when you no longer want it to appear in the representation of an VMware Cloud Director API objects, or when you want to replace it with a new service link.

When you retrieve the list of service links associated with a service, the response is a `QueryResultRecords` element in which each service link is represented as a `ServiceLinkRecord` element. The value of the `href` attribute of a `ServiceLinkRecord` is a URL you can use to retrieve or delete the service link.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the service.

This request retrieves the XML representation of the service created in Register an Extension Service:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

2 Examine the response to find the `Link` for listing service links.

   This `Link` has the following form:

```
<vcloud:Link
      rel="down:serviceLinks"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/44/links" />
```

3 Make a GET request to the link described in Step 2.

4 Examine the response to find the `ServiceLinkRecord` that represents the service link to delete.

5 Make a DELETE request to the URL in the href attribute value of that `ServiceLinkRecord`.

Example: Delete a Service Link

Start by getting the service's list of service links.

Request:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryResultRecords
   total="3"
   pageSize="25"
   page="1"
   name="serviceLink"
   type="application/vnd.vmware.vcloud.query.records+xml"
   ... >
   ...
   <ServiceLinkRecord
      service="https://vcloud.example.com/api/admin/extension/service/45"
      resourceType="application/vnd.vmware.vcloud.vApp+xml"
      rel="down"
      mimeType="application/vnd.vmware.vcloud.backupList+xml"
      linkHref="{baseUri}api/vApp/vapp-{resourceId}/backups"
      href="https://vcloud.example.com/api/admin/extension/service/link/01ca..." />
   <ServiceLinkRecord
      service="https://vcloud.example.com/api/admin/extension/service/45"
      resourceType="application/vnd.vmware.vcloud.org+xml"
      rel="down"
      mimeType="application/vnd.vmware.vcloud.backupList+xml"
      linkHref="{baseUri}org/{resourceId}"
      href="https://vcloud.example.com/api/admin/extension/service/link/a624..." />
   <ServiceLinkRecord
      service="https://vcloud.example.com/api/admin/extension/service/45"
      resourceType="application/vnd.vmware.vcloud.vApp+xml"
      rel="backup"
      mimeType="application/vnd.vmware.vcloud.createBackupParams+xml"
      linkHref="{baseUri}vApp/vapp-{resourceId}/action/backup"
      href="https://vcloud.example.com/api/admin/extension/service/link/f62e..." />
</QueryResultRecords>
```

Note Link `id` values are truncated in this example.

Using this information, find the `ServiceLinkRecord` that represents the service link you want to delete, and make a `DELETE` request to that URL.

```
DELETE https://vcloud.example.com/api/admin/extension/service/link/f62e...
```

Service-Specific Tasks and Events

An extension service can create `Task` objects in a VMware Cloud Director organization, and can post events to the organization's event stream.

Tasks and events are created in the context of an organization. Each task or event is associated with exactly one user, who must be a system administrator or a member of the organization in which the task or event is created. Tasks and events can also have an owner, which is a reference to the subject of the task or event (for example, an object being created or updated by a task).

Task and Event Workflow

VMware Cloud Director native services typically create `Task` objects to track the progress of asynchronous events. These objects are returned to clients as `Task` elements, which can be embedded in a container that represents an object under construction, or simply returned in `Task` form.

Tasks are also reported in the organization's event stream. A service can also add its own events to this stream, in addition to the ones added as a side-effect of creating a task.

Extension services act as VMware Cloud Director API clients when creating tasks and events, even if those tasks and events are created to track service-specific objects or operations. After a task is created in VMware Cloud Director, an extension service can use its AMQP connection to VMware Cloud Director to return a `Task` as the response, or part of the response, to a client request.

Localizing Task and Event Message Content

Message strings included in tasks and events can be localized. See Localization Framework for Extension Services.

Create or Update a Service-Specific Task

When a user requests an asynchronous operation from an extension service, the service can create a task object and add it to an organization's tasks list.

Every VMware Cloud Director organization has a tasks list and accepts requests to add a task to the list. When a client requests an asynchronous operation from a service, the service starts to process the request and also POSTs a `Task` element to the organization's `tasksList` URL. VMware Cloud Director adds information such as an `id` and `startTime` to the `Task`, places it on the organization's `TasksList`, creates an event in the organization's event stream, and returns the `Task` to the service. The service can then send the `Task`, as an AMQP message, to VMware Cloud Director, which sends it as a response to the client that made the original request.

Note Because of the diversity of sources from which an extension service can draw references to the `User`, `Owner`, and `Organization` elements of a `Task`, it may not always be possible for every client to resolve such references. For example, if a service creates an object in an organization of which you are not a member, you will not be able to resolve the reference to the object in the `Owner` element of the `Task`.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the organization in which you want to create the `Task`.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26
```

2 Examine the response to locate the `Link` element that contains the URL for adding tasks to the organization's tasks list.

This element has a `rel` attribute value of `task:create` and a `type` attribute value of `application/vnd.vmware.vcloud.task+xml`, as shown here:

```
<Link
   rel="task:create"
   type="application/vnd.vmware.vcloud.task+xml"
   href="https://vcloud.example.com/api/tasksList/26" />
```

3 Create a `Task` element that specifies the details of the task.

4 POST the `Task` element to the organization's `tasksList` URL.

Results

The server creates a task object and adds it to the organization's tasks list, and returns the representation of the object to the service. To return the XML representation of the task object to the client that made the original request, the service must create a JSON representation of the `Task` and return it to VMware Cloud Director AMQP service.

Example: Add a Task to an Organization's Tasks List

Request:

```
POST https://vcloud.example.com/api/tasksList/26
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Task
   xmlns="http://www.vmware.com/vcloud/v1.5"
   status="running"
   serviceNamespace="org.example.vcd.backup"
   type="application/vnd.vmware.vcloud.task+xml"
   operation="Backup in progress for virtual machine with id 7b91b053-2b..."
   operationName="backupInProgress"
   name="task">
 <Owner
      type="application/vnd.vmware.vcloud.org+xml"
      name="Finance"
      href="https://vcloud.example.com/api/org/26"
      id="26" />
   <User
      type="application/vnd.vmware.admin.user+xml"
      name="bob"
      href="https://vcloud.example.com/api/admin/user/39" />
   <Progress>10</Progress>
</Task>
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
   href="https://vcloud.example.com/api/task/604"
   ...
   operationName="backupInProgress"
   ... >
   ...
</Task>
```

Create a Service-Specific Event

An extension service can request that VMware Cloud Director add an event message to the event stream of an organization.

The system always creates an event message when a service posts a `Task` to an organization's tasks list. To create additional event messages, a service can POST an `Event` element to an organization's `events` URL.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the organization in which you want to create the event.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/org/26
```

2 Examine the response to locate the `Link` element that contains the URL for adding events to the organization's events stream.

This element has a `rel` attribute value of `event:create` and a `type` attribute value of `application/vnd.vmware.vcloud.event+xml`, as the following example shows:

```
<Link
   rel="event:create"
   type="application/vnd.vmware.vcloud.event"
   href="https://vcloud.example.com/api/admin/org/26/events" />
```

3 Create an `Event` element that specifies the details of the task.

4 POST the `Event` element to the organization's `events` URL.

Example: Add an Event to an Organization's Event Stream

Request:

```
POST https://vcloud.example.com/api/admin/org/26/events
Content-Type: application/vnd.vmware.admin.event+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<Event
   success="true"
   serviceNamespace="org.example.vcd.backup"
   type="backupComplete">
   <Owner
      type="application/vnd.vmware.vcloud.organization+xml"
      name="Finance"
      href="https://vcloud.example.com/api/admin/org/26" />
   <User
      type="application/vnd.vmware.admin.user+xml"
      name="bob"
      href="https://vcloud.example.com/api/admin/user/39" />
</Event>
```

Response:

```
204 No Content
```

Authorization Framework for Extension Service Operations

All requests to extension services must be authenticated through the VMware Cloud Director API. Extension services can participate in VMware Cloud Director API REST authorization by controlling access to their objects and operations through new or existing rights and roles.

An extension service that does not enable the use of VMware Cloud Director REST authorization implicitly grants permission for all users to perform all operations that the service uses. A service can use the native VMware Cloud Director REST authorization model by taking the following steps:

1 Define resource classes that represent references to service-specific object types.

2 Define resource class actions that specify the actions that are implemented for those object types.

3 Define ACL rules specifying the rights required to perform an operation on objects of a specific type.

Participation in the Authorization Framework

To participate in the authorization framework, a service must include an `AuthorizationEnabled` element with a value of `true` in its registration request.

```
<vmext:AuthorizationEnabled>true</vmext:AuthorizationEnabled>
```

It must also define at least one resource class, specify at least one action for that class, and define an ACL rule that constrains use of the action on the class.

Resource Classes and Actions

A service uses the following constructs to define the objects, operations, and permissions that constitute its authorization model.

Resource Classes

Set of rules for creating references to service-specific objects. Like other object references in the VMware Cloud Director API, resource classes are a `Link` element that specifies the MIME `type` of the resource and includes an `href` (URL) that can be used to retrieve the resource. The rules include a MIME type, a URL pattern, and a template for creating an `id` attribute value in URN form.

Resource Class Actions

Combination of a URL pattern that specifies a resource class and an HTTP method that implements an action on a resource of that class. The action uses the specified method in a request to a URL that matches the specified pattern.

ACL Rules

Specifies the rights that an organization or user have to an operation defined as a resource class action.

Service Resource

A member of a resource class distinguished by a specific `id`. If an extension service needs to define a resource class action or an ACL rule that applies to a specific resource, the service must create it as a `ServiceResource` and give it a UUID or other unique identifier.

Querying for Organization and User Rights

The VMware Cloud Director API query service implements several queries that return a list of rights that a specified user or organization is granted. A user can make a request that specifies one or more entity references and returns a summary of user rights to the specified entities.

Create an Extension Service Resource Class

To configure your extension service to provide access control for the objects it creates, define a resource class for each of its object types.

A `ResourceClass` element contains the information needed to construct a URL that a client can use to access the resource in a specific context. It must contain the following child elements:

`MimeType`

The MIME content type of all instances of the resource class.

`UrlTemplate`

The value of `href` attribute value for resources of this class. This can be any URI, and can include the variables `{baseUri}` and `{resourceId}`. When constructing the `href` value, VMware Cloud Director replaces `{baseUri}` with the VMware Cloud Director Director REST API base URL, and replaces `{resourceId}` with the UUID portion of the `id` attribute value of the resource.

`Nid`

The Namespace Identifier for resources of this type, as specified in http://www.ietf.org/rfc/rfc2141.txt.

`UrnPattern`

The Namespace Specific String for resources of this type, as specified in http://www.ietf.org/rfc/rfc2141.txt. You can provide a string or a named regular expression, where `(?<id>)` matches the resource identifier.

Prerequisites

This operation is restricted to system administrators.

Procedure

1. Retrieve the XML representation of the service.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/admin/extension/service/45
   ```

2. Examine the response to find the `Link` for adding resource classes.

   This `Link` has the following form:

   ```
   <vcloud:Link
         rel="add"
         type="application/vnd.vmware.admin.resourceClass+xml"
         href="https://vcloud.example.com/api/admin/extension/service/45/resourceclasses" />
   ```

3. Construct a `ResourceClass` element.

   See the request portion of Create an Extension Service Resource Class for information about the contents of this element.

4. POST the `ResourceClass` element to the URL described in Step 2.

Example: Create an Extension Service Resource Class

This request defines a resource class named `Backup`.

- The `MimeType` is specified using the standard form for `vnd` type names.
- The `UrlTemplate` uses the `{baseUri}` and `{resourceId}` variables, and could expand to a URL like `https://vcloud.example.com/backup/27`
- The `Nid` and `UrnPattern` elements provide rules for constructing an URN of the form:

  ```
  urn:vcloud:backup:id
  ```

  as shown in the response.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/resourceclasses
Content-type:application/vnd.vmware.admin.resourceClass+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:ResourceClass
   name="Backup"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
   <vmext:MimeType>application/vnd.vmware.vcloud.backup+xml</vmext:MimeType>
   <vmext:UrlTemplate>{baseUri}backup/{resourceId}</vmext:UrlTemplate>
   <vmext:Nid>vcloud</vmext:Nid>
   <vmext:UrnPattern>^backup(?&lt;id&gt;[0-9]*)</vmext:UrnPattern>
</vmext:ResourceClass>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.resourceClass+xml
...
<vmext:ResourceClass
   name="Backup"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   id="urn:vcloud:backup:83"
   type="application/vnd.vmware.admin.resourceClass+xml"
   href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83"
   ...>
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83" />
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.service+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45" />
   <vcloud:Link
      rel="down:resourceClassActions"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83/
resourceclassactions" />
   <vcloud:Link
      rel="down:serviceResources"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83/
serviceresources" />
   <vmext:MimeType>BackupType+xml</vmext:MimeType>
   <vmext:UrlTemplate>{baseUri}backup/{resourceId}</vmext:UrlTemplate>
   <vmext:Nid>nidBackup</vmext:Nid>
   <vmext:UrnPattern>^myNssBackup(?&lt;id&gt;[0-9]*)</vmext:UrnPattern>
</vmext:ResourceClass>
```

Define an Action for a Resource Class

After you define a resource class, you can specify the actions that are permitted on resources of that class.

A `ResourceClassAction` object defines an HTTP method that is allowed on a specific `UrlPattern`. The `UrlPattern` can be any of the following URL forms:

- An explicit URL like`/backup/restore/vm/27`. A `UrlPattern` of this form defines an action for a resource that has a specific URL.
- A URL that contains a regular expression, like `/backup/restore/vm[-,a-z,0-9]*`. A `UrlPattern` of this form defines an action for any resource in the class that matches the regular expression.

- A URL that contains a service resource `id`, which is expressed as `<id>`. In a `UrlPattern`, the delimiters must be written as the XML entities `&lt;` and `&gt;`. The `id` can stand alone, as in `/backup/restore/vm/&lt;id&gt;`, or appear as part of a regular expression like `/backup/restore/vm/?&lt;id&gt;[-,a-z,0-9]*)`. A `UrlPattern` of this form targets a specific `ServiceResource`, which must the service must define and register.

Prerequisites

This operation is restricted to system administrators.

Procedure

1. Retrieve the XML representation of the service.

   Use a request like this one:

   ```
   GET https://vcloud.example.com/api/admin/extension/service/45
   ```

2. Examine the response to find the `Link` for specifying resource class actions.

   This `Link` has the following form:

   ```
   <vcloud:Link
         rel="add"
         type="application/vnd.vmware.admin.resourceClassAction+xml"
         href="https://vcloud.example.com/api/admin/extension/service/45/
   resourceclassactions" />
   ```

3. Construct a `ResourceClassAction` element.

   For information about the contents of this element, see the request portion of Define an Action for a Resource Class.

4. POST the `ResourceClassAction` element to the URL described in Step 2.

Example: Define an Action for a Resource Class

This example defines a resource class action for a GET request to a `UrlPattern` that could match URLs including `https://vcloud.example.com/api/backup/27` or `https://vcloud.example.com/api/backup/vm-27-backup-2013-04-25T01:17:00.000Z`.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/resourceclassactions
Content-type:application/vnd.vmware.admin.resourceClassAction+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:ResourceClassAction
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="Read backups">
   <vmext:HttpMethod>GET</vmext:HttpMethod>
   <vmext:UrlPattern>/api/backup/[-,a-z,0-9]*)</vmext:UrlPattern>
</vmext:ResourceClassAction>
```

The response is a ResourceClassAction element that includes information derived from the contents you POSTed , along with a set of Link elements that you can use to remove the ResourceClassAction or add ACL rules that control access to the resource class through the action.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.resourceClassAction+xml
...
<vmext:ResourceClassAction
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="Read backups"
   id="urn:vcloud:resourceClassAction:268"
   type="application/vnd.vmware.admin.resourceClassAction+xml"
   href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268"
   ... >
   <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268" />
   <vcloud:Link
      rel="up"
      type="application/vnd.vmware.admin.resourceClass+xml"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclass/83" />
   <vcloud:Link
      rel="down:aclRules"
      type="application/vnd.vmware.vcloud.query.records+xml"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268/
aclrules" />
   <vmext:HttpMethod>GET</vmext:HttpMethod>
   <vmext:UrlPattern>/api/backup/[-,a-g,0-9]*)</vmext:UrlPattern>
</vmext:ResourceClassAction>
```

Define an ACL Rule for a Resource Class Action

Permission to execute an extension service operation is controlled by an AclRule contained in the ResourceClassAction.

An ACL rule specifies the access controls that apply to a ResourceClassAction. Access controls can be defined for any of the following principals:

- an individual user
- a member of a specified organization
- any user whose role includes a specific right
- any resource defined by the service that created the ACL rule

Rights for specific entity types are specified in the following container elements:

ServiceResourceAccess

This specification is optional.

OrganizationAccess

Access for the organizations. This specification is required.

PrincipalAccess

Access control for users, or for any role that includes a specified right. This specification is required.

If the Access element in any of these containers has the value Entity, the container must also include an Entity element that provides a reference to a resource entity, organization, user, or right.

Table 12-3. ACL Rules

Container Element	Access	Comments
ServiceResourceAccess	Shared	The action is authorized for all resources in this resource class
	Entity	The action is authorized for the service resource referenced in the Entity element in this container.
OrganizationAccess	Shared	The action is authorized for all members of the organization that owns the resource.
	Published	The action is authorized for all members of any organization in the cloud.
	Entity	The action is authorized for members of the organization referenced in the Entity element in this container.
PrincipalAccess	Shared	The action is authorized for all users
	Entity	The action is authorized for the User referenced in the Entity element in this container, or for any role that includes the Right referenced in the Entity element in this container.

A ResourceClassAction can include an arbitrary number of AclRule elements. The action is permitted if the user or resource attempting the action matches any rule.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the resource class action.

This request retrieves the XML representation of the resource class action created in Define an Action for a Resource Class:

```
GET https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268
```

2 Examine the response to find the `Link` for specifying ACL rules for the resource class action.

   This `Link` has the following form:

```
<vcloud:Link
      rel="add"
      type="application/vnd.vmware.vcloud.query.aclRule+xml"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268/
aclrules" />
```

3 Construct an `AclRule` element.

   See the request portion of Define an ACL Rule for a Resource Class Action for information about the contents of this element.

4 POST the `AclRule` element to the URL described in Step 2.

Example: Define an ACL Rule for a Resource Class Action

This example adds an ACL rule to the resource class action created in Define an Action for a Resource Class. The rule specifies that all members of a specific organization who have a role that includes a specific right can execute the action.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268/aclrules
Content-type: application/vnd.vmware.admin.aclRule+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:AclRule
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   name="ACL rule for read backups">
   <Description>Only users in org/26 who have right/2 can read backups</Description>
   <vmext:ServiceResourceAccess>
      <vmext:Access>Shared</vmext:Access>
   </vmext:ServiceResourceAccess>
   <vmext:OrganizationAccess>
      <vmext:Access>Entity</vmext:Access>
      <vmext:Entity
         xsi:type="vcloud:ResourceReferenceType"
         type="application/vnd.vmware.admin.organization+xml"
         href="https://vcloud.example.com/api/admin/org/26" />
   </vmext:OrganizationAccess>
   <vmext:PrincipalAccess>
      <vmext:Access>Entity</vmext:Access>
      <vmext:Entity
         xsi:type="vcloud:ResourceReferenceType"
         type="application/vnd.vmware.admin.right+xml"
         href="https://vcloud.example.com/api/admin/right/2" />
   </vmext:PrincipalAccess>
</vmext:AclRule>
```

The response contains information supplied in the request, along with several `Link` elements created by the server.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.aclrule+xml
...
<vmext:AclRule
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   name="ACL rule for read backups"
   id="urn:vcloud:aclRule:5"
   type="application/vnd.vmware.admin.aclRule+xml"
   href="https://vcloud.example.com/api/admin/extension/service/aclrule/5">
   <Description>Only users in org/26 who have right/2 can read backups</
Description><vcloud:Link
    <vcloud:Link
      rel="remove"
      href="https://vcloud.example.com/api/admin/extension/service/resourceclassaction/268" />
   ...
</vmext:AclRule>
```

Create a Service-Specific Right

A service can create rights that apply to its operations. You can add these rights to existing roles or new roles.

In the VMware Cloud Director API, a right is simply a name that a service attaches to a privilege. When a service specifies an ACL rule for a resource class action, the rule can reference a right. A user who is assigned a role that includes the right is authorized to take the specified action.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the service.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

2 Examine the response to find the `Link` for adding or listing service-specific rights

This `Link` has the following form:

```
<vcloud:Link
      rel="rights"
      type="application/vnd.vmware.admin.rights+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/rights" />
```

3. Construct a Right element.

   For information about the contents of this element, see the request portion of Create a Service-Specific Right.

4. POST the Right element to the URL described in Step 2.

Example: Create a Service-Specific Right

This request creates a right named DeleteBackup. The name attribute and Category element are required, and can have any string value. Include a BundleKey if any messages associated with the right appear in a localization bundle.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/rights
Content-type:application/vnd.vmware.admin.right+xml
<?xml version="1.0" encoding="UTF-8"?>
<Right
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="DeleteBackup">
   <Description>Right to remove a backup object</Description>
   <Category>VcdBackup</Category>
   <BundleKey>BackupBundle</BundleKey>
</Right>
```

The response is a Right element that includes information derived from the contents you POSTed. The service namespace name is prepended to the name of the right.

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.right+xml
...
<Right
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="{org.example.vcd.backup}:DeleteBackup"
   id="urn:vcloud:right:99"
   type="application/vnd.vmware.admin.right+xml"
   href="https://vcloud.example.com/api/admin/right/99"
   ... >
   <Description>Right to remove a backup object</Description>
   <Category>VcdBackup</Category>
   <BundleKey>BackupBundle</BundleKey>
</Right>
```

Localization Framework for Extension Services

Extension service developers can provide localized content for service-specific tasks and events by creating and uploading a localization bundle.

An extension service localization bundle is a file in `zip` format that contains one or more properties files. Each file in the bundle has a name that indicates the locale to which it applies, and contains an arbitrary number of *key*=*value* pairs, where the *key* is the value of an attribute of a service-specific task operation or event, and the *value* is a localized string to display in log messages posted by the service.

Upload or Update a Localization Bundle

Each service provides a link that an administrator can use to upload a new or modified localization bundle for the service.

A localization bundle is a file in ZIP format that contains one or more localized message files that your service uses. Each line in one of these files provides localized text that replaces the text that your service posts in the values of certain attributes for service-specific events and tasks. See Message File Content.

Prerequisites

- This operation is restricted to system administrators.
- Create a localization bundle.

Procedure

1. Create a localization bundle.
2. Find the `localizationbundles` URL for your service.
   a. Retrieve the XML representation of the service.
   b. Examine the representation for a `Link` of the following form:

   ```
   <vcloud:Link
      rel="bundle:upload"
      type="application/vnd.vmware.admin.bundleUploadParams+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/
   localizationbundles" />
   ```

   The `localizationbundles` URL is the value of the href attribute of this link.
3. Create a `BundleUploadParams` element that specifies the size of the bundle and the service namespace of the service.
4. POST the `BundleUploadParams` element to the `localizationbundles` URL of your service.

Example: Upload a Localization Bundle

This example uploads a localization bundle for the service created in Register an Extension Service. The initial request specifies the size of the ZIP file in bytes and the name of the service namespace.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/localizationbundles
Content-type: application/vnd.vmware.admin.bundleUploadParams+xml
...
<?xml version="1.0" encoding="UTF-8"?>
<vmext:BundleUploadParams
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5"
   fileSize="537"
   serviceNamespace="org.example.vcd.backup">
</vmext:BundleUploadParams>
```

The response body includes an `uploadLocation` URL. You can use a procedure similar to the one in Uploading File Data to upload the ZIP file to this location.

Response:

```
<vmext:BundleUploadSocket
   uploadLocation="https://vcloud.example.com/transfer/6f3b.../file"
   xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5"
   xmlns:vcloud="http://www.vmware.com/vcloud/v1.5">
</vmext:BundleUploadSocket>
```

To update a bundle, follow the same procedure using a new bundle that contains updated message files. When you upload a new localization bundle for a service that already has one, new *key*=*value* pairs are added, and the values of existing keys are updated.

Message File Content

Message files provide localized text for certain attribute values in `Task` and `Event` objects.

Each file in a localization bundle can include an arbitrary number of lines. Each line must have the following form, where *key* is any of the values that the service might assign to one of its service-specific `Task` or `Event` elements and *value* is the localized string to display.

```
key=value
```

The *value* string can contain parameters that provide the `resourceId`, `resourceName`, and `resourceType` of the subject of the `Task` or `Event`.

The following restrictions apply to each message file in a localization bundle:

- File contents must be encoded in the UTF-8 character set.
- *key* length cannot exceed 2000 UTF-8 characters.
- *value* length cannot exceed 2000 UTF-8 characters.
- File size cannot exceed 10MB.
- File name must be the locale code for the language used in the *value* strings. For example, the file containing English text must be named `en_US`. The file containing French text must be named `fr_FR`.

Whenever a localizable attribute appears in a log message, VMware Cloud Director takes the following steps to find text to display:

1 If the service has a localization bundle, open the file in that bundle whose name corresponds to the current client locale and display the value as it appears in the file.

2 If the service has a localization bundle but no file exists in that bundle whose name corresponds to the current client locale, open the file in that bundle named en_US and display the value as it appears in the file. The default locale for VMware Cloud Director is en_US.

3 Otherwise, display a predefined string.

A Task or Event that a service posts can also include a passthrough *key*=*value* pair that is always displayed as posted by the service, regardless of the current client locale or the presence or absence of a localization bundle.

Message File Keys and Parameters

Message file contents (*key*=*value* pairs) apply to all Task and Event objects that the service creates. If a Task and an Event both use the same *key*, the same message appears for both.

Table 12-4. Localization Keys for Service-Specific Tasks

Attribute Values Matched for *key*	Available Parameters
operationName	**resourceId** The value of the id attribute of the Owner of the Task. **resourceName** The value of the name attribute of the Owner of the Task. **resourceType** The value of the type attribute of the Owner of the Task.
Owner:type	None
serviceNamespace	None
operation (passthrough)	None

To create a message that appears only when a Task has status="running", append the string _PROGRESS to the *key*.

Table 12-5. Localization Keys for Service-Specific Events

Attribute Values Matched for *key*	Available Parameters

`type`	**`resourceId`**
	The value of the `id` attribute of the `Owner` of the `Event`.
	`resourceName`
	The value of the `name` attribute of the `Owner` of the `Event`.
	`resourceType`
	The value of the `type` attribute of the `Owner` of the `Event`.

`Owner:type`	None
`serviceNamespace`	None
`typeFull` (passthrough)	None

To create a message that appears only for a failed `Event` (one where `success="false"`), prepend the string `FAILED.` to the key.

Example: Example Message File Content

The following lines are appropriate in the `en_US` message file for a service that meets the following conditions:

- Its `Namespace` is registered as `org.example.vcd.backup`.
- It defined a `Task` whose `operationName` attribute can have a value of `backupInProgress`.
- It defined an `Event` whose `type` attribute can have a value of `backupComplete`.

```
org.example.vcd.backup=vCloud Backup Service
backupInProgress=Backup in progress for ${resourceName} ({resourceType}) with id: {resourceId}
backupInProgress_PROGRESS=Backup in progress for ${resourceName} ({resourceType}) with id:
{resourceId}
backupComplete=Backup complete for entity {resourceName} ({resourceType}) with id:
{resourceId}
FAILED.backupComplete=Backup failed for entity {resourceName} ({resourceType}) with id:
{resourceId}
```

If the localization bundle for this service contained a file named `fr_FR` that included the following line, the `Task` posted in Add a Task to an Organization's Tasks List returns this localized value for the `operationName` attribute when the client locale is set to `fr_FR`. The passthrough value for `operation` is not localized.

```
backupInProgress_PROGRESS=Sauvegarde en cours pour entity {resourceName} ({resourceType})
avec id: {resourceId}
```

Request:

```
GET https://vcloud.example.com/api/task/604
```

Response:

```
200 OK
Content-Type: application/vnd.vmware.vcloud.task+xml
...
<Task
   ...
   operation="Backup in progress for virtual machine with id 7b91b053-2b..."
   operationName="Sauvegarde en cours pour entity Finance (application/
vnd.vmware.vcloud.org+xml) avec id 26 "
   ... >
```

REST APIs for Extension Services

A simple extension service does not need a REST API. You can define a service-specific REST API entrypoint and one or more schema definition files.

An extension service that does not require request or response bodies other than those that the VMware Cloud Director API defines, `Task`, for example, can simply define the URL patterns that constitute its API filters and the service links that implement its operations.

A service that defines its own request or response bodies must also specify a URL to which clients can direct requests. The service must specify locations of the files, such as XML schema definition (XSD) files, to which its clients require access.

Create an API Filter for an Extension Service

When you register an extension service with VMware Cloud Director, you specify one or more API filters, which are URL patterns or MIME content types that the VMware Cloud Director REST service should treat as extension requests. You cannot update the API filter for a registered service, but you can replace it with a new one.

An API filter can be either a URL pattern, typically in the form of a regular expression, or a content type, typically in the form of a MIME content-type string. Requests whose URL matches the specified `UrlPattern` are sent to the service that has registered the filter. An API filter that specifies `ResponseContentType` is applied only to responses whose `Content-type` attribute has a value that matches the specified `ResponseContentType`. An extension service that receives such a response must return it, after making any service-specific modifications, to the AMQP service as a JSON message, so that it can be returned to the VMware Cloud Director client that made the request.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the service.

This request retrieves the XML representation of the service created in Register an Extension Service:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

2 Examine the response to find the `Link` for adding API filters

This `Link` has the following form:

```
<vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.apiFilter+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/apifilters" />
```

3 Construct an `ApiFilter` element.

This `ApiFilter` overwrites any existing `ApiFilter` defined by the service. See the request portion of Create an API Filter for an Extension Service for information about the contents of this element.

4 POST the `ApiFilter` element to the URL described in Step 2.

Example: Create an API Filter for an Extension Service

This request adds a new `UrlPattern` to set of patterns defined in the request portion of Register an Extension Service. The request replaces the existing set of filter expressions with a new one that includes the original set and one additional expression.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/apifilters
Content-type:application/vnd.vmware.admin.apiFilter+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:ApiFilter>
   <vmext:UrlPattern>(/api/org/.*/backups)|(/api/vApp/vapp-.*/backups)|(/api/vApp/vapp-.*/
action/backup)|(/api/backup/.*) | (/api/vApp/vapp-.*/action/recoverBackup)</vmext:UrlPattern>
</vmext:ApiFilter>
```

Response:

```
201 Created
Content-Type: application/vnd.vmware.admin.resourceClass+xml
...
<vmext:ApiFilter>
   <vmext:UrlPattern>(/api/org/.*/backups)|(/api/vApp/vapp-.*/backups)|(/api/vApp/vapp-.*/
action/backup)|(/api/backup/.*) | (/api/vApp/vapp-.*/action/recoverBackup)</vmext:UrlPattern>
</vmext:ApiFilter>
```

Create or Update an Extension Service API Definition

You can define an API for an extension service when you register the service. You can also create or update the service API definition later.

An API definition for an extension service includes an API entry point (A URL at which a client can send requests to the API) and a set of file descriptors, each of which consists of a description and a reference to a schema definition file.

An `ApiDefinition` element must contain the following child elements:

`EntryPoint`

The URL to which a client can send requests to the service API. This can be any URL, and can include the variable `{baseUri}`, which represents the VMware Cloud Director REST API base URL.

`Namespace`

The service namespace. See Register an Extension Service

`Files`

One or more references to schema definition files. The references must be accessible to VMware Cloud Director.

Prerequisites

This operation is restricted to system administrators.

Procedure

1 Retrieve the XML representation of the service.

Use a request like this one:

```
GET https://vcloud.example.com/api/admin/extension/service/45
```

2 Examine the response to find the `Link` for adding API definitions.

This `Link` has the following form:

```
<vcloud:Link
      rel="add"
      type="application/vnd.vmware.admin.apiDefinition+xml"
      href="https://vcloud.example.com/api/admin/extension/service/45/apidefinitions" />
```

3 Construct an `ApiDefinition` element.

For information about the contents of this element, see the request portion of Create an Extension Service API Definition.

4 POST the `ApiDefinition` element to the URL described in Step 2.

Example: Create an Extension Service API Definition

This request defines an API for a backup service. The definition includes two `FileDescriptor` elements that reference files available on the vendor's public Website. The entrypoint for requests to the service is the VMware Cloud Director API login URL.

Request:

```
POST https://vcloud.example.com/api/admin/extension/service/45/apidefinitions
Content-type:application/vnd.vmware.admin.apiDefinition+xml
<?xml version="1.0" encoding="UTF-8"?>
<vmext:ApiDefinition
   name="Backup service version 5.1">
   <Description>Backup service API</Description>
   <vmext:EntryPoint>{baseUri}/login</vmext:EntryPoint>
   <Namespace>org.example.vcd.backup</Namespace>
   <vmext:Files>
      <vmext:FileDescriptor>
         <vmext:Description>Master schema definition file.</vmext:Description>
         <vmext:File
            href="http://example.com/backp/schema/v5.1/master.xsd" />
      </vmext:FileDescriptor>
      <vmext:FileDescriptor>
         <vmext:Description>Schema definition file for backup devices.</vmext:Description>
         <vmext:File
            href="http://example.com/vcdBackup/schema/v5.1/devices.xsd" />
      </vmext:FileDescriptor>
   </vmext:Files>
</vmext:ApiDefinition>
```

Response:

```
201 Created
Content-type:application/vnd.vmware.admin.apiDefinition+xml
...
<vmext:ApiDefinition
   name="Backup service version 5.1">
   ...
</vmext:ApiDefinition>
```

Extension Service AMQP Message Format

VMware Cloud Director uses the system AMQP service to communicate with extension services. Messages to and from an extension service are formatted as JSON objects.

When it receives an extension request, the VMware Cloud Director REST service creates a message and sends it to the system AMQP service, specifying the exchange and routing key registered by the extension service. The extension service retrieves the message from a queue bound to the exchange it registered, processes the request, and returns a response to the common reply exchange.

AMQP Message Headers

Each message from VMware Cloud Director to a service includes both standard and custom AMQP headers.

Table 12-6. Extension Service AMQP Message Headers

Header	Value
`correlationId`	A standard AMQP header that provides a unique identifier for the message. The extension must supply the same correlationId in the corresponding response.
`reply-to`	A standard AMQP header specifying the value that the extension must use as the `routingKey` in the response.
`messageType`	A custom AMQP header. One of: **`ProcessHttpRequest`** Indicates that this message is a forwarded request. **`ProcessHttpResponse`** Indicates that this message is a forwarded response.
`replyToExchange`	A custom AMQP header. The name of the AMQP exchange to which the extension should publish its response.

Property Names and Values

A request message contains all of the following `name=value` pairs.

Table 12-7. Extension Service AMQP Request Message Property Names and Values

Name	Value
`method`	The HTTP method (GET, PUT, POST, DELETE) used to make the request
`id`	The unique id of this message
`scheme`	The scheme (HTTP or HTTPS) specified in the request URL
`protocol`	The protocol used to make the request
`headers`	The request headers represented as a map of *name*:*value* pairs encoded as a JSON object in the form: `name:value,name:value,...`
`queryString`	The entire query string, or `null` if the request did not include a query string.
`localPort`	The local port to which the request was sent
`remoteAddr`	The IP address of the requesting machine
`remotePort`	The remote port from which the request was sent

Table 12-7. Extension Service AMQP Request Message Property Names and Values (continued)

Name	Value
`localAddr`	The IP address to which the request was sent
`request`	Always `true` in request messages
`requestURI`	The request URL, without any query string it might have included
`parameters`	always null
`user`	The `id` of the VMware Cloud Director user who made the request
`org`	The `id` of the VMware Cloud Director organization to which the requesting user belongs
`rights`	A comma-separated list of id values for the VMware Cloud Director rights assigned to the requesting user.

The `parameters`, `user`, `org`, and `rights` properties provide the security context for the request, and are formatted as a separate JSON object, as shown in AMQP Message Format

A response message contains all of the following `name=value` pairs.

Table 12-8. Extension Service AMQP Response Message Property Names and Values

Name	Value
`id`	The unique id of this message
`headers`	A comma-separated list of request headers in the form: `name:value,name:value,...`
`statusCode`	The HTTP status code to return to the requester
`body`	A base64-encoded response body
`request`	Always `false` in response messages

Example: AMQP Message Format

Assume an extension service that includes an API filter of the following form:

```
<vmext:ApiFilter>
   <vmext:UrlPattern>/api/org/.*</vmext:UrlPattern>
</vmext:ApiFilter>
```

When VMware Cloud Director receives a request like this one:

```
GET https://10.23.6.168:8443/api/org/a93c9db9-7471-3192-8d09-a8f7eeda85f9
```

it creates the following message and places it on the service's exchange.

```
[
  {
    "method":"GET",
    "id":"32d5b9ec-5eef-4aa3-9375-b054018b0e30",
    "scheme":"https",
    "protocol":"HTTP/1.1",
    "headers":{"Cookie":"...", "User-Agent":"...", ...},
    "queryString":null,
    "localPort":8443,
    "remoteAddr":"10.23.6.168",
    "remotePort":60576,
    "localAddr":"10.100.1.40",
    "request":true,
    "requestUri":"/api/org/a93c9db9-7471-3192-8d09-a8f7eeda85f9"
  },
  {
    "parameters":null,
    "user":"urn:vcloud:user:8cdd352f-f831-4712-a1a3-9e061687c5c6",
    "org":"urn:vcloud:org:a93c9db9-7471-3192-8d09-a8f7eeda85f9",
    "rights":["urn:vcloud:right:0b8c8cd2-5af9-32ad-a0bd-dc356503a552",...]
  },
  null
]
```

The service returns a response containing a base64-encoded body.

```
[
  {
    "id":"32d5b9ec-5eef-4aa3-9375-b054018b0e30",
    "headers":{"Date":"...", "Content-Type":"application/
vnd.vmware.vcloud.org+xml;version=2.0"},
    "statusCode":200,
    "body":"base64-encoded-body",
    "request":false,
  }
]
```

XML Representations in the VMware Cloud Director API

13

The VMware Cloud Director API represents objects in a cloud as XML documents in which object properties are contained in elements and attributes that have typed values and an explicit object hierarchy defined by an XML schema.

Client programs of RESTful Web services must be able to request object representations from the server, parse the server's responses to extract the information they contain, and compose requests that, in many cases, are based on the information extracted from a response. Developers of such clients must understand the structure of each representation that might be part of a request or response, and any requirements that the network protocol (HTTP) places on client-server interaction.

XML Schemas

Each VMware Cloud Director API object is defined in an XML schema document. Schema files and reference information about all elements, types, operations, and queries is included in the *VMware Cloud Director API Schema Reference*. See About the Schema Reference.

VMware Cloud Director uses a validating XML parser that requires elements in XML documents to agree in order and number with the schema. Required elements must appear in request bodies. All elements that appear in request bodies must appear in the order established by the schema, and with content that conforms to the type constraint specified in the schema. Default values, where defined, are supplied for elements that are empty. See XML Namespace Identifiers.

All VMware Cloud Director API requests are processed in the `http://www.vmware.com/vcloud/v1.5` XML namespace. VMware Cloud Director API XML namespace information appears in the values of the `xsi:schemaLocation` and `xmlns` attributes in a response document.

```
xmlns="http://www.vmware.com/vcloud/v1.5"
xsi:schemaLocation="https://vcloud.example.com/api/v1.5/schema/master.xsd"
```

Other XML namespace identifiers may also be required in request bodies. See XML Namespace Identifiers.

API Versions

The VMware Cloud Director XML namespace (`http://www.vmware.com/vcloud/v1.5`) defines elements and attributes for all supported versions of the VMware Cloud Director API. Treatment of version-specific elements and attributes in requests is controlled by the value of the `version` attribute in the `Accept` header. For example, this `Accept` header specifies that the request body is presumed to be valid for VMware Cloud Director API version 20.0 and a version 20.0 response is expected:

```
Accept: application/*;version=20.0
```

Requests are validated against the elements and attributes defined in the specified version. Responses are filtered to remove elements and attributes that are not defined in the specified version. In general, client requests can access objects defined by any version of the VMware Cloud Director API that is less than or equal to the version specified in the `Accept` header. Exceptions to this rule are mentioned in the VMware Cloud Director *Release Notes*. The *VMware Cloud Director API Schema Reference* indicates the deprecation status of elements and attributes, and also indicates when each element or attribute was added to the API. See About the Schema Reference.

To discover the API versions that a server supports, a client can make an unauthenticated GET request to a well-known URL on the server. See Retrieve the Login URL and List of Supported API Versions.

Date and Time Values

Values of type `xs:dateTime` are always interpreted as UTC if a timezone has not been explicitly specified.

Length Limits on Element and Attribute String Values

String values for the `name` attribute and the `Description` and `ComputerName` elements have length limitations that depend on the object to which they are attached.

Table 13-1. Length Limits on Element and Attribute String Values

Object	Element or Attribute Name	Maximum Length in Characters
`Catalog`	`name`	128
`Catalog`	`Description`	256
`EdgeGateway`	`name`	35
`Media`	`name`	128
`Media`	`Description`	256
`VApp`	`name`	128

Table 13-1. Length Limits on Element and Attribute String Values (continued)

Object	Element or Attribute Name	Maximum Length in Characters
VApp	Description	256
VAppTemplate	name	128
VAppTemplate	Description	256
Vdc	name	256
Vdc	Description	256
Vm	name	128
Vm	ComputerName	15 on Windows, 63 on all other platforms

A VM name cannot contain any special characters. See VMware Knowledge Base article https://kb.vmware.com/kb/2046088.

Extensibility

The VMware Cloud Director API provides complete programmatic access to the VMware Cloud Director Extension Services facility. See VMware Cloud Director Extension Services.

In addition, there is a more general extensibility mechanism, VCloudExtension, that clients are free to use. VCloudExtensibleType is an abstract type that all complex types defined in the VMware Cloud Director API namespace extend. It can contain an arbitrary number of elements and attributes, and provides a way for you to add custom attributes and elements to any type.

The VCloudExtension element has an attribute named required that specifies how clients and servers proceed when they see an unknown extension. All VCloudExtension elements are assumed to require a server that understands them. The required attribute is optional, but if omitted is assumed to be present with a value of true. This extensibility mechanism allows new servers to extend the XML representations native to the VMware Cloud Director API without requiring existing clients to understand those extensions.

A client might encounter a VCloudExtension element in any response. If the element declares required=”true” and the client does not know how to interpret the contents of the element, the client can ignore it, but it must include the VCloudExtension in any request to modify the element that contains it. A server must return a failure when a request includes a VCloudExtension element that declares required=”true” but the server does not understand the extension. For more information about VCloudExtension, see the schema reference.

Read the following topics next:

- XML Namespace Identifiers
- Common VMware Cloud Director API Attributes
- Retrieve an Object as an Entity

XML Namespace Identifiers

Elements used as request or response bodies contain a set of attributes that enable XML validation. The body of a PUT or POST request must contain all XML namespace identifiers required to validate the elements it contains. A response body typically includes all the XML namespace identifiers that the server used to validate it, in addition to other attributes that specify the schema locations searched during validation.

The VMware Cloud Director API uses these XML namespace identifier attributes and prefixes.

Table 13-2. XML Namespace Identifiers in the VMware Cloud Director API

Name	Value	Requirement
xmlns	http://www.vmware.com/vcloud/v1.5	Required in all request bodies.
xmlns:vmext	http://www.vmware.com/vcloud/extension/v1.5	Required in request bodies that include elements from the vSphere platform extensions.
xmlns:ve	http://www.vmware.com/schema/ovfenv	Required in request bodies that include an `ovf:Environment`element.
xmlns:ovf	http://schemas.dmtf.org/ovf/envelope/1	Required in request bodies that include elements defined in OVF schema http://schemas.dmtf.org/ovf/envelope/1/dsp8023.xsd.
xmlns:rasd	http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData	Required in request bodies that include elements defined in OVF schema CIM_ResourceAllocationSettingData.xsd.
xmlns:oe	http://schemas.dmtf.org/ovf/environment/1	Required in request bodies that include elements defined in OVF schema dsp8027_1.1.0.xsd.
xmlns:vssd	http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData	Not required in request bodies.
xsi:schemaLocation	An installation-dependent schema location search path. See http://www.w3.org/TR/xmlschema-0/.	Not required in request bodies.
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance	Not required in request bodies.

XML Namespace Prefixes in Request and Response Bodies

When a request or response includes elements from multiple XML namespaces, each element name is prefixed with a namespace identifier. Unless all elements in a request or response originate in the same XML namespace, these prefixes are required in request bodies, and are always included in response bodies.

The examples omit XML namespace identifiers from most responses. The following fragment shows how some of them appear in a typical response body.

```
<VApp
   xmlns="http://www.vmware.com/vcloud/v1.5"
   ...
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5 https://vcloud.example.com/api/
schema/v1.5/master.xsd">
   ...>
   ...
</VApp>
```

Common VMware Cloud Director API Attributes

Most VMware Cloud Director API objects have a number of common attributes. Except the `name` attribute, these common attributes are not required in request bodies, and are ignored if included. All of them are included in response bodies.

Object Name

Every object requires a `name` attribute. The string value of this attribute is included in all object references, and can be used as the display name for the object. The value of `name` must be unique within a given scope.

Table 13-3. Requirements for Unique Object Names

Object Type	Name Scope
ProviderVdc	Cloud
Org	Cloud
Vdc	Organization
Catalog	Organization
CatalogItem	Catalog
vAppTemplate	None
vApp	Organization
Vm	vApp
Media	Catalog
Disk	None
Network	Container (Organization VDC, vApp, or cloud)

Object Identifier, Type, and Reference

These attributes are common to all object representations.

id

The object identifier, expressed in URN format. The value of the `id` attribute uniquely identifies the object, persists for the life of the object, and is never reused. The `id` attribute value is intended to provide a context-free identifier that can be used with the VMware Cloud Director API `entityResolver` (see Retrieve an Object as an Entity).

type

The object type, specified as a MIME content type.

href

An object reference, expressed in URL format. This reference includes the object identifier portion of the `id` attribute value, and supplies additional information, including the current location of the object when accessed in a specific view. Although URLs have a well-known syntax and a well-understood interpretation, a client should treat each `href` as an opaque string. The rules that govern how the server constructs `href` strings might change in future releases.

Object Creation Status

Objects such as `VAppTemplate`, `VApp`, and `Vm`, that extend the `ResourceEntity` type have a `status` attribute whose value indicates the state of the object. In this table, YES indicates that a status value is allowed for the object listed in the column header. The `status` value for a `VAppTemplate` or `VApp`, which contain `Vm` objects that each have a `status` attribute of their own, is computed from the status of the contained objects. When returned in an XML representation, `status` has a numeric value. When returned by the query service, it has a string value.

Table 13-4. `status` Attribute Values for VAppTemplate, VApp, Vm, and Media Objects

Numeric Value	String Value	Description	vAppTemplate	App	Vm	Media
-1	FAILED_CREATION	The object could not be created.	YES	YES	YES	YES
0	UNRESOLVED	The object is unresolved.	YES	YES	YES	YES
1	RESOLVED	The object is resolved.	YES	YES	YES	YES
2	DEPLOYED	The object is deployed.	No	No	No	No
3	SUSPENDED	The object is suspended.	No	YES	YES	No
4	POWERED_ON	The object is powered on.	No	YES	YES	No

Table 13-4. `status` Attribute Values for VAppTemplate, VApp, Vm, and Media Objects (continued)

Numeric Value	String Value	Description	vAppTemplate	App	Vm	Media
5	WAITING_FOR_INPUT	The object is waiting for user input.	No	YES	YES	No
6	UNKNOWN	The object is in an unknown state.	YES	YES	YES	No
7	UNRECOGNIZED	The object is in an unrecognized state.	YES	YES	YES	No
8	POWERED_OFF	The object is resolved and powered off.	YES	YES	YES	No
9	INCONSISTENT_STATE	The object is in an inconsistent state.	No	YES	YES	No
10	MIXED	Children do not all have the same status. Also, a vApp might have one or more VMs in the `PARTIALLY_POWERED_OFF` or `PARTIALLY_SUSPENDED` status.	YES	YES	No	No
11	DESCRIPTOR_PENDING	Upload initiated, OVF descriptor pending.	YES	No	No	No
12	COPYING_CONTENTS	Upload initiated, copying contents.	YES	No	No	No
13	DISK_CONTENTS_PENDING	Upload initiated , disk contents pending.	YES	No	No	No
14	QUARANTINED	Upload has been quarantined.	YES	No	No	No
15	QUARANTINE_EXPIRED	Upload quarantine period has expired.	YES	No	No	No
16	REJECTED	Upload has been rejected.	YES	No	No	No
17	TRANSFER_TIMEOUT	Upload transfer session timed out.	YES	No	No	YES
18	VAPP_UNDEPLOYED	The vApp is resolved and undeployed.	YES	No	No	No
19	VAPP_PARTIALLY_DEPLOYED	The vApp is resolved and partially deployed.	YES	No	No	No

Table 13-4. `status` Attribute Values for VAppTemplate, VApp, Vm, and Media Objects (continued)

Numeric Value	String Value	Description	vAppTemplate	App	Vm	Media
20	PARTIALLY_POWERED_OFF	The VM is deployed and powered off. (Applicable to VMware Cloud Director API version 36.1)	No	No	Yes	No
21	PARTIALLY_SUSPENDED	The VM is deployed and suspended. (Applicable to VMware Cloud Director API version 36.1 and later)	No	No	Yes	No

VDC objects have their own set of `status` values and mappings.

Table 13-5. `Status` Attribute Values for VDC Objects

Numeric Value	String Value	Description
-1	FAILED_CREATION	The VDC could not be created.
0	NOT_READY	The VDC is not ready for use
1	READY	The VDC is ready for use
2	UNKNOWN	The VDC status cannot be retrieved
3	UNRECOGNIZED	The VDC status cannot be mapped to a known state.

Retrieve an Object as an Entity

You can use the VMware Cloud Director API entity resolver with an object's `id` attribute value to retrieve a context-free reference to the object.

Every first-class object that the VMware Cloud Director API defines includes an `id` attribute whose value is the object identifier expressed in URN format. The value of the `id` attribute uniquely identifies the object, persists for the life of the object, and is never reused.

You can append the value of the `id` attribute to the VMware Cloud Director API `entityResolver` URL to retrieve a context-free representation of the underlying object as an `Entity` element. The `Entity` includes a `Link` element for each currently valid reference to the object identified by the `id` specified in the request.

Prerequisites

Verify that you are logged in to the VMware Cloud Director API.

Procedure

1 Retrieve the current `Session` object to get the `entityResolver` URL.

Use a request like this one:

```
GET https://vcloud.example.com/api/session
```

The response is a `Session` element, which includes a link to the `entityResolver`.

```
<Session ... >
  ...
    <Link
      rel="entityResolver"
      type="application/vnd.vmware.vcloud.entity+xml"
      href="https://vcloud.example.com/api/entity/" />
</Session>
```

2 Retrieve the object whose `id` you want to resolve and find the value of its `id` attribute.

See the request portion of Using the entityResolver URL.

3 Append the value of the object's `id` attribute to the `entityResolver` URL.

4 Make a GET request to the URL you created in Step 3

See the request portion of Using the entityResolver URL.

Example: Using the entityResolver URL

This example retrieves the organization object shown in Object id, type, and href Attributes as an `Entity`.

Request:

```
GET https://vcloud.example.com/api/entity/urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4
```

This response includes two `Link` elements, each of which provides a valid `href` to the object identified by the `id` specified in the request.

Response:

```
<Entity
   xmlns="http://www.vmware.com/vcloud/v1.5"
   id="urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   name="urn:vcloud:org:72fe715c-5f6a-407f-bbb2-bf465915b5f4"
   type="application/vnd.vmware.vcloud.entity+xml"
   href="https://vcloud.example.com/api/entity/urn:vcloud:org:72fe715c-5f6a-407f-bbb2-
bf465915b5f4"
   ...>
   <Link
      rel="alternate"
      type="application/vnd.vmware.vcloud.organization+xml"
      href="https://vcloud.example.com/api/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4"/>
   <Link
```

```
      rel="alternate"
      type="application/vnd.vmware.vcloud.admin.organization+xml"
      href="https://vcloud.example.com/api/admin/org/72fe715c-5f6a-407f-bbb2-bf465915b5f4"/>
</Entity>
```