VMware vCloud® Architecture Toolkit Implementation Examples

Version 3.0

September 2012

vmware®

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

Contents

1. Overview 9
1.1 Implementation Examples Structure 9
1.2 vCloud Suite Components 10
2. vCloud Cell Design Examples 12
2.1 Load Balanced Cell Configuration 12
2.2 Secure Certificates 17
3. Organization Virtual Datacenter Examples 29
3.1 Pay-As-You-Go Allocation Model 29
3.2 Reservation Pool Model 32
3.3 Allocation Pool Model 35
3.4 Service Provider Performance Offerings 39
4. Networking Examples 43
4.1 vApp Load Balancing with vCloud Networking and Security Edge 43
4.2 Static Routing 51
4.3 vCloud Networking and Security Edge Gateway Setup 56
4.4 Public vCloud External Network 68
4.5 VXLAN ORG Network for Disaster Recovery 72
4.6 VCDNI-Backed Organization Network 88
4.7 VLAN ORG Network 92
5. Storage Design Examples 98
5.1 vApp Snapshot 98
5.2 Storage DRS with vCloud Director 102
6. Catalog Design Example 111
6.1 vCloud Public Catalog 111
7. vCloud Security Examples 118
7.1 Single Sign-On (SSO) – Provider 118
7.2 Single Sign-On (SSO) – Consumer 128
8. vCloud Management and Monitoring Examples 139
8.1 vCenter Operations Manager 139
8.2 AMQP Messages 174
8.3 AMQP Blocking Tasks 178

List of Figures

Figure 1. VMware vCloud Director Abstraction Layer 11
Figure 2. HTTPS and Console Proxy Connections 16
Figure 3. Requesting, Configuring, Obtaining and Installing an SSL Certificate from QuoVadis 17
Figure 4. Pay-As-You-Go Settings 31
Figure 5. Reservation Pool Settings 34
Figure 6. Configure Allocation Pool Screen 37
Figure 7. Graphical Summary of Components 44
Figure 8. Routing Example Logical Architecture 52
Figure 9. Service Provider External Network Example 69
Figure 10. vSphere Port Group Configuration 70
Figure 11. vCloud External Networks 70
Figure 12. vcd-ext-101 External Network Configuration 70
Figure 13. Network Specification Properties 71
Figure 14. Example Logical Architecture 73
Figure 15. Example Physical Architecture 73
Figure 16. vCloud Director Network Configuration 74
Figure 17. Removed External Network 76
Figure 18. Test 3 End Result 77
Figure 19. VCDNI Network Pool Example Configuration 88
Figure 20. VCDNI-Backed Network Pool Creation 89
Figure 21. Organization Virtual Datacenter Network – IP Address Settings 91
Figure 22. VLAN Network Pool Example Configuration 93
Figure 23. VLAN-Backed Network Pool Settings 94
Figure 24. Organization Virtual Datacenter Network – IP Address Settings 96
Figure 25. Network Pool Corresponding vSphere Port Groups 96
Figure 26. Two-Tier vApp 98
Figure 27. Two-Tier vApp as Seen in vCenter 99
Figure 28. The Process of Creating a vApp Snapshot 99
Figure 29. Snapshot Options 100
Figure 30. Snapshot Creation 100
Figure 31. Creation of an Additional Snapshot for the Web Virtual Machine 101
Figure 32. The Consumer Rolls Back the DB Virtual Machine 101

Figure 33. vSphere Tasks When Reverting a Snapshot 102
Figure 34. Overview of Storage Profiles Architecture 103
Figure 35. vApp Deployment Storage Profile 110
Figure 36. Cloud Provider SSO Logical Architecture 119
Figure 37. Single Sign-On Authentication Workflow 127
Figure 38. Single Sign-On (SSO) Between a Client and Multiple Back End Services 129
Figure 39. SSO Solution-to-Solution Authentication 130
Figure 40. Task Execution on Behalf of a User 131
Figure 41. Single Sign-On for Long-Lived Tasks 132
Figure 42. Consumer Logical Single Sign-On Deployment Architecture 132
Figure 43. Consumer Workflow Detail 137
Figure 44. vCenter Operations Manager vApp Components 140
Figure 45. Hyperic Configuration 141
Figure 46. vCenter Operations Manager and Hyperic Integration 142
Figure 47. Generic Cluster CPU Scoreboard 146
Figure 48. Management Cluster CPU Core Utilization Heat Map 148
Figure 49. CPU Resource Cluster Dashboard 149
Figure 50. Resource Cluster CPU Scoreboard Widget 150
Figure 51. Resource Cluster Physical CPU Core Heat Map Widget 150
Figure 52. Resource Cluster CPU Metric Graph Widget 150
Figure 53. Cluster Health Widget 151
Figure 54. Cluster Memory Dashboard 151
Figure 55. Cluster Memory Scoreboard Widget 152
Figure 56. Cluster Memory Heat Map Widget 153
Figure 57. Cluster Memory Metric Graph Widget 154
Figure 58. Storage Dashboard 155
Figure 59. Cluster Disk Scoreboard Widget 156
Figure 60. Cluster Disk Capacity Scoreboard Widget 157
Figure 61. Cluster Storage Metric Graph Widget 158
Figure 62. Network Dashboard Widget 159
Figure 63. Outbound and Inbound Packet Rate Metric Graph 160
Figure 64. Completed Network Scoreboard Widget 161
Figure 65. Physical NIC Heat Map Widget 161
Figure 66. Network Performance Metric Graph Widget 163
Figure 67. Health Status Dashboard 164

Figure 68. Cluster Health Bar 165
Figure 69. Alerts Widget 166
Figure 70. Alerts Information Detail 166
Figure 71. Capacity Remaining Scoreboard Dashboard 168
Figure 72. Management Cluster Dashboard 169
Figure 73. Completed Widget 170
Figure 74. Completed Widget xxxxx 171
Figure 75. AMQP Blocking Task Architecture 179
Figure 76. Enable a Task for Blocking in vCloud Director 181
Figure 77. vCloud Director AMQP Configuration 182

List of Tables

Table 1. Example Layout 9
Table 2. vCloud Components 10
Table 3. Company2 Pay-As-You-Go Organization Settings 30
Table 4. Company2 Reservation Pool Organization Settings 33
Table 5. Company2 Allocation Pool Organization Settings 36
Table 6. Company1 Pay-As-You-Go Offering 40
Table 7. Company1 Reservation Pool Offering 40
Table 8. Company1 Allocation Pool Offering 41
Table 9. Network Device Information 43
Table 10. Network Device Information 53
Table 11. Sample NAT Rules 74
Table 12. Existing versus Revised vCloud DR Process 78
Table 13. vCloud Director Networks 90
Table 14. vCloud Director Networks 95
Table 15. vCenter Operations Enterprise Port Access Requirements 144
Table 16. vCenter Operations Manager Data Source URLs 144
Table 17. Generic Scoreboard Widget Configuration 149
Table 18. Cluster Memory Metric Graph Settings 153
Table 19. Cluster Storage Usage Widget Settings 156
Table 20. Cluster Disk Capacity Scoreboard Settings 157
Table 21. RabbitMQ Server Exchange Configuration 175
Table 22. RabbitMQ Server Queue Configuration 175
Table 23. RabbitMQ Server Exchange Configuration 179
Table 24. AMQP Queue Configuration 180

1. Overview

VMware vCloud Architecture Toolkit Implementation Examples provides detailed examples for using the VMware vCloud® Suite. The examples highlight key design decisions associated with implementing an Infrastructure as a Service (IaaS) solution. Each example is a module that can provide a baseline or component of an overall IaaS design. The examples refer to the fictitious companies "Company1" and "Company2."

The examples are intended to serve as a reference for architects and engineers, and assume a level of familiarity with VMware products in the VMware vCloud Suite, including VMware vSphere®, VMware vCenter™, and VMware vCloud Director™ (VCD).

1.1 Implementation Examples Structure

Use the implementation examples as a reference for a specific technology or feature in the vCloud Suite to quickly find and research an area of interest. Each example is organized as shown in the following table.

Table 1. Example Layout

Section	Notes
x.x <Example Name>	
Deployment Models	Deployment models for this technology example (private, public, hybrid, all).
Example Components	The required software components and versions. For example: vSphere 5.1, vCloud Director 5.1.
x.x.1 Background	Background about a specific technology example and an overview that describes how it can be used.
x.x.2 Example	An example of the use of the technology or feature for a specific use case.
x.x.3 Design Implications	Information to consider when using the technology or feature.

For each example, see the VMware product installation and administration guides for additional information.

1.2 vCloud Suite Components

The following table describes the components that comprise the VMware vCloud Suite.

Table 2. vCloud Components

vCloud Component	Description
VMware vCloud Director vCloud API	Layer of software that abstracts virtual resources and exposes vCloud components to consumers. Includes: • vCloud Director Server (also referred to as a cell). • vCloud Director Database. • VMware vCloud API, used to manage vCloud objects programmatically.
VMware vSphere	Virtualization platform providing abstraction of physical infrastructure layer for vCloud. Includes: • VMware ESXi™ hosts. • VMware vCenter™ Server. • vCenter Server database.
VMware vCloud Networking and Security	Decouples network and security from the underlying physical network hardware through software-defined networking and security. Includes: • VXLAN support. • vCloud Networking and Security Edge Gateway. • App and Data Security. • Manager.
VMware vCenter Operations Management Suite	Provides predictive capacity and performance planning, compliance and configuration management, dynamic resource metering, cost modeling, and report generation using the following components: • vCenter Operations Manager. • vCenter Configuration Manager. • vCenter Infrastructure Navigator. • vCenter Chargeback Manager.
vFabric Application Director	Part of the Cloud Application Platform family of products that provide automated provisioning of application infrastructure.
VMware vCenter Orchestrator™	Enables the automation of provisioning and operational tasks across VMware and third-party applications using an open and flexible plug-in architecture.

VMware vCloud Connector	vSphere Client plug-in that enables users to connect to vSphere-based or vCloud Director-based clouds and manage them through a single interface.

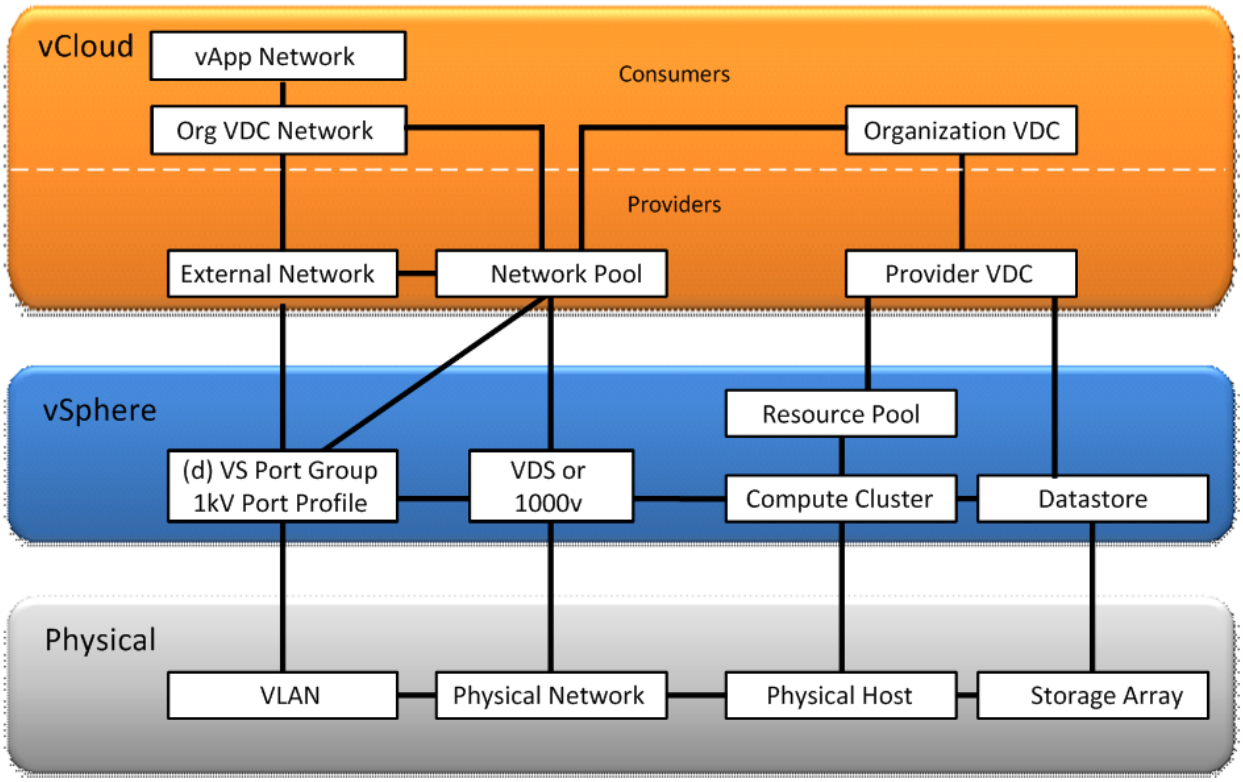
1.2.1 VMware vCloud Director

VMware vCloud Director further abstracts the virtualized resources presented by vSphere by providing the following logical constructs that map to vSphere logical resources:

- *Organization* – A logical object that provides a security and policy boundary. Organizations are the main method of establishing multitenancy and typically represent a business unit, project, or customer in a private vCloud environment.
- *Virtual datacenter* – Deployment environments in which virtual machines run.
- *Organization virtual datacenter* – An organization's allocated portion of provider virtual datacenter resources, including CPU, RAM, and storage.
- *Provider virtual datacenter* – vSphere resource groupings of compute, storage, and network resources that power organization virtual datacenters.

Figure 1 shows the vCloud Director abstraction layer in relation to vSphere and physical resources.

Figure 1. VMware vCloud Director Abstraction Layer



2. vCloud Cell Design Examples

2.1 Load Balanced Cell Configuration

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

2.1.1 Background

A vCloud Director cell is comprised of Red Hat Enterprise Linux (supported versions as documented in the VMware Compatibility Guide) with installed and configured VMware vCloud Director binary files. This server becomes the portal for the vCloud Director environment, for both administrators (providers) and users (consumers) of the vCloud.

This example architecture covers the following considerations:

- Shared Transfer Space.
- Load Balancer setup.
- SSL offload for certificates and SSL to cells.
- Load balancer health check rules.
- External URL settings in vCloud Director.

2.1.2 Example

This example demonstrates how to set up load balanced vCloud Director cells. The following prerequisites align with and are explained in the setup considerations.

2.1.2.1. Prerequisites

- More than one vCloud Director supported cell server (Red Hat Enterprise Linux).
- Supported network latency (in this example, < 2ms RTT between vCloud Director cells). The environment must behave as though it is a single site (that is, low latency).
- Location and password of the keystore file that includes the SSL certificates for this server.
- Shared transfer storage space, mapped on each vCloud Director cell.
- Shared database instance.
- Network Time Protocol configured on each cell.
- Network Load Balancer device, and Console and Proxy Health Monitors.
- vCloud Director “Public Addresses” configuration.

2.1.2.2. Setup Considerations

- More than one vCloud Director cell is required. Multiple vCloud Director cells are desirable for redundancy, and to provide a sufficient number of console sessions.
  - If the first vCloud Director cell is lagging due to too many console connections, adding more load balanced cells offloads the workload between the cells by providing more console connections.
  - If the first cell becomes unavailable, users can no longer use the vCloud Director portal or console sessions. Having multiple, load balanced cells enables user sessions to be re-established across other cells.

  In this example, the customer wants to gain vCloud Director cell redundancy by load balancing the cells. There is no need to get more console access connections because of the limited number of users accessing the vCloud Director portal or vCloud Director backspace-presented virtual machine consoles.
- Network latency between the cells must be low enough to perform as though the cells reside in the same datacenter (LAN speed). Cells must have low latency between each other and also between themselves and the shared database. This provides proper communication and updates between the cells, along with the cell to database traffic.

  The customer is hosting all of the vCloud Director cells within two adjacent racks in a datacenter. This configuration has less than a 2ms round trip time (RTT) between the cells and corresponding services (database, vSphere, and so on).
- Certificates are generated when a vCloud Director cell is installed. See the *vCloud Director Installation Guide* for information about creating the proper certificates.

  During the creation of the certificates, the HTTP and Console IP addresses are identified, along with the keystore location and password. Record this information. The default location is `/opt/keystore/certificates.ks`.

  The customer wants to create an untrusted (self-signed) certificate using the following command. This command creates an untrusted certificate for the HTTP service in a keystore file named `certificates.ks`:

  ```
  keytool -keystore certificates.ks -storetype JCEKS -storepass passwd
  -genkey -keyalg RSA -alias http
  ```

The following table contains the customer provided information used to populate the certificate as prompted during the certificate creation from the preceding command.

Certificate Prompted Question	Customer Answer
What is your first and last name?	Mycloud.example.com
What is the name of your organization unit?	IT
What is the name of your organization?	Example Company Name
What is the name of your city or locality?	Tucson
What is the name of your state or province?	Arizona
What is the two-letter country code for this unit?	US

The following command adds an untrusted certificate for the console proxy to the keystore file created in the previous step:

```
keytool -keystore certificates.ks -storetype JCEKS -storepass passwd
-genkey -keyalg RSA -alias consoleproxy
```

- For the transfer service, each server must mount an NFS or other shared storage volume at `$VCLOUD_HOME/data/transfer`, which is typically `/opt/vmware/vcloud-director/data/transfer`. This volume must have write permission for root.

  Note: In some lab and POC environments, it may be acceptable, though not recommended, to share a local NFS mount from one cell server to other cell servers. Keep in mind that if this NFS server becomes unavailable, functions and services using this share will become unavailable and copy operations in progress will fail.

- Database connection information and other reusable responses you supplied during the configuration are preserved in a file located at `/opt/vmware/vcloud-director/etc/responses.properties` on this server. Copy this file to each additional cell server before the configuration of the vCloud Director cell server software. This file is referenced during the initiation of the installation using this command:

  ```
  installation-file -r path-to-response-file
  ```

- The Network Time Protocol (NTP) must be properly configured on each vCloud Director cell. vCloud Director cells with improperly configured NTP settings can have trouble connecting to ESXi hosts, the database instance, vCloud Networking and Security Manager, and other cells. If there is one cell in the group without the correct time and a user connects through this cell, actions will not complete correctly and many errors will occur. This is due to the time stamps from that cell trying to access the load balanced environment (other cells) and the shared database with time stamps that are not matching the times of the improperly configured cell.

- A network load balancer, whether physical or virtual, must be configured to proxy the HTTP and console connections for the vCloud Director environment. This section also includes setting up the health monitoring on the load balancer for the vCloud Director environment.

To configure the HTTP Portal connections

1. Copy the SSL certificate to the load balancer for the HTTP public URL (that is, https://mycloud.example.com).
2. Set up the Health Monitor on the Load Balancer (F5 Load Balancer in this customer example) using the following URL:

   https://<Cell_Hostname>/cloud/server_status

3. Each node can have a hostname-based cert node1.example.com, node2.example.com, and so forth.

To configure the Console Proxy connections

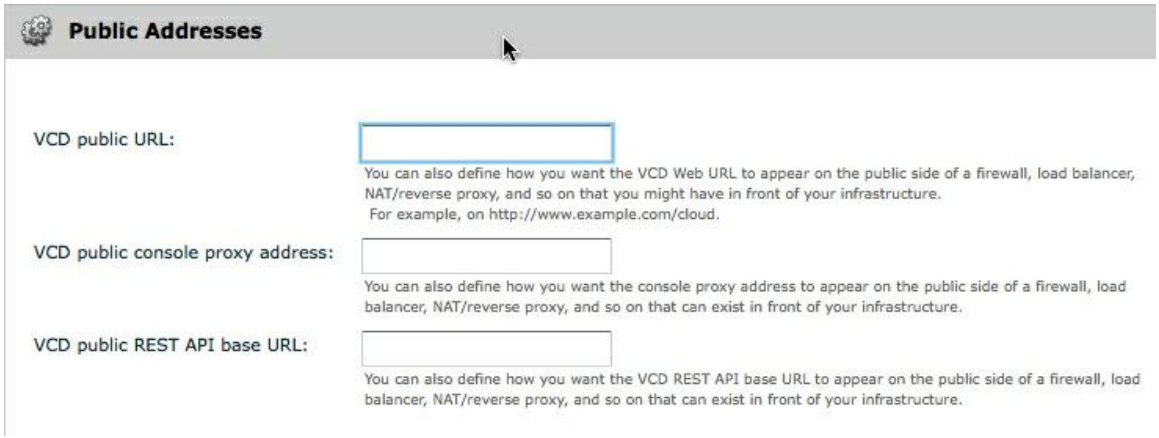
1. Configure HTTPS pass-through for the console proxy connections. Each node should have the same host name in the certificate when it is generated (that is, http://mycloud.example.com). For more details, see the *vCloud Director Installation and Configuration Guide*. Console connections are load balanced. Because the certificates are the same for each cell console, they must be passed through the load balancer. Otherwise, each cell would need a unique certificate in the load balancer and a certificate mismatch error would occur.
2. Set up the Health Monitor on the Load Balancer (F5 in this customer example) using the following URL:

   ```
   https:///sdk/vimServiceVersions.xml
   ```

3. Configure SSL persistence for the vCloud Director load balancer connections.
4. In the vCloud Director system administration section, there are fields for the vCloud Director public URL, console proxy address, and REST API base URL. If these fields are not updated with the load balancer public IP addresses/URLs, replies from the cell will reflect that of the individual cell and not that of the public URL information.

For example (from the *vCloud Director Installation and Configuration Guide)*:

- When you create an organization, its organization URL includes the public web URL instead of the HTTP service IP address. vCloud Director also modifies the organization URLs of existing organizations.
- Remote console session tickets sent to the HTTP service IP address return the public console proxy address.
- XML responses from the REST API include the base URL and the transfer service uses the base URL as the upload target.

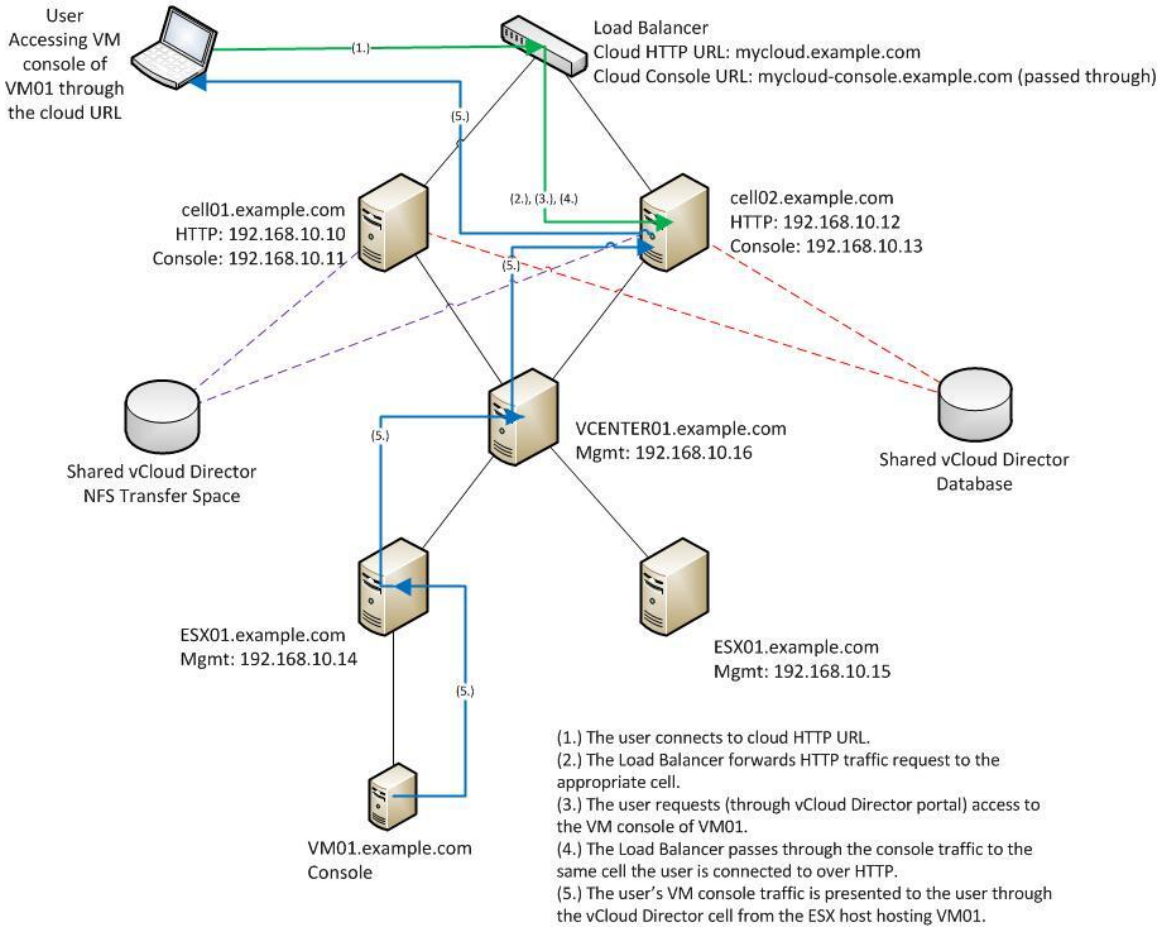


Public Addresses Field	Example Data
VCD public URL	http://mycloud.example.com
VCD public console proxy address	mycloud-console.example.com
VCD public REST API base URL	mycloud.example.com

- The **VCD public URL** requires HTTPS. This address is also set in the load balancer and DNS so that users can access the vCloud Director portal using this public URL.
- The **VCD public console proxy address** does not use HTTPS. This URL is also in DNS and on the load balancer in order to direct users to the appropriate cell for virtual machine console access.
- The **VCD public REST API base URL** is for users to access the load balance environment using the REST API.

The following figure outlines traffic communication between the cells and ESXi hosts.

Figure 2. HTTPS and Console Proxy Connections



To access the console of a virtual machine, vCloud director uses the console proxy IP address of the VCD cell server to connect directly (through the load balancer, in this case) to the ESXi host and attach the user to the console of the target virtual machine.

When a user connects to the vCloud Director portal, a different URL is used (also using the load balancer in this case) and the user connects through to the HTTPS vCloud Director portal hosted on the vCloud Director cells. All actions and commands executed within the vCloud Director portal are sent directly (transparent to the end user) to the relevant vCenter Server or, in some cases, directly to the ESXi host.

All vCloud Director cells in the same vCloud must access the same vCloud Director database. Each vCloud Director cell must also be configured for the same NFS mount point, which is used as vCloud Director transfer space.

2.2 Secure Certificates

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

2.2.1 Background

Security is a critical component of a successful vCloud deployment. Before you can install and run VMware vCloud Director you must implement certificates and key management for secure access and authentication to the vCloud Director server. The following example shows how to implement security features designed to safeguard data, keep out intruders, and allow access to legitimate users.

Using the SSL/TLS protocol in the vCloud environment provides secure communication between the end-tenant (client) and vCloud Director cell (server). Providing secure communication requires:

- Confidentiality and privacy of communication.
- Message integrity and hashing.
- Authentication.

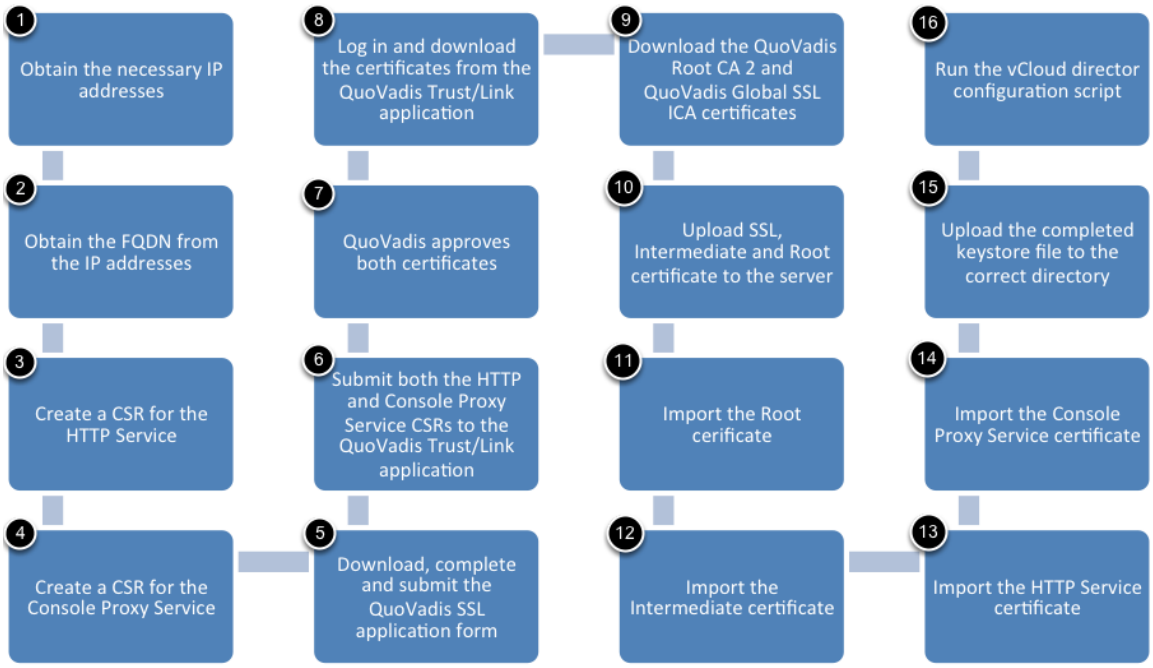
2.2.2 Example

Regardless of whether you are a private, hybrid or public vCloud provider, securing communication between end tenants of the vCloud portal and the vCloud Director infrastructure usually requires implementing SSL Certificates from a trusted Certificate Authority (CA). The following example uses the QuoVadis CA to issue SSL certificates for vCloud Director.

This process flow shown in Figure 3 includes procedures for requesting, configuring, obtaining, and installing an SSL certificate from QuoVadis.

QuoVadis is used for this example, but any trusted CA can be used.

Figure 3. Requesting, Configuring, Obtaining and Installing an SSL Certificate from QuoVadis



2.2.2.1. Prerequisites for Creating the Required Certificate Signing Requests

Before creating a Certificate Signing Request (CSR), you must know the IP address and FQDN (fully qualified domain name) of your servers.

To list server information and change to the keytool directory

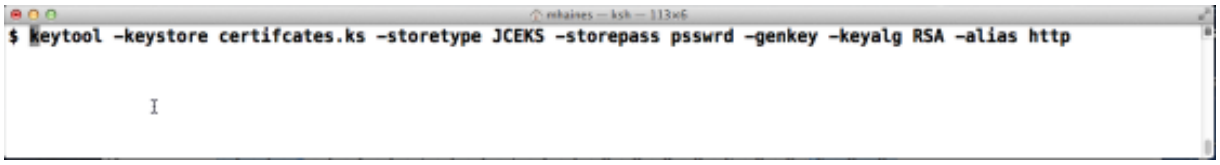
1. From the vCloud Director cell, run `ifconfig` (8) to list the IP addresses for this server. Record the two IP addresses that correspond to the vCloud Director HTTP service interface and the Console Proxy Service interface.
2. To obtain the FQDNs, use the command `nslookup ipaddress`. Record these FQDNs, which are needed for the HTTP server and Console Proxy service SSL certificates.
3. Change your directory to `/opt/vmware/vcloud-director/jre/bin/keytool`.

`keytool` is installed along with vCloud Director by default. Alternatively, you can use `keytool` on another computer that has a Java version 6 runtime environment installed, and then import the created Java `Keystore` file onto your vCloud Director server. This example assumes that you are using the `keytool` installed on the vCloud Director server.

2.2.2.2. Part I – Creating the CSR for the HTTP Service

To create the CSR for the HTTP service

1. After you have navigated to the `keytool` directory, run the command shown in the following screenshot.



   Note: You can change the values for variables, but any changes that you make must be used throughout the entire process. For example, if the keystore name is changed from `certificates.ks` to `mysslcertificate.ks`, then you must continue to use `mysslcertificate.ks` in place of `certificates.ks`.

2. When prompted, type your first and last name.
3. When prompted, type the FQDN (fully qualified domain name) to use for the HTTP service certificate.
4. Type the following answers when prompted:

   What is your first and last name? [Unknown]: **mycloud.mydomain.com**

   What is the name of your organizational unit? [Unknown]: **MyCompanyDivision**

   What is the name of your organization? [Unknown]: **MyCompanyLegalName**

   What is the name of your City or Locality? [Unknown]: **CityOfMyCompany**

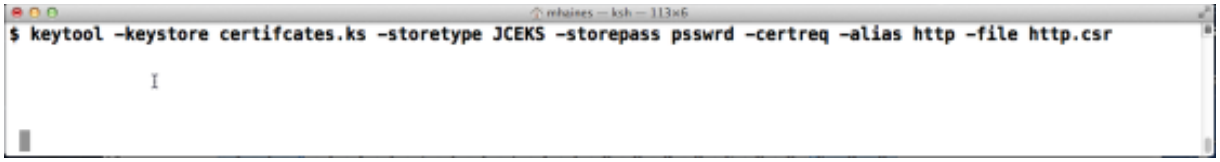
   What is the name of your State or Province? [Unknown]: **StateOfMyCompany**

5. Type **yes** to continue when `keytool` summarizes your entries.

   ```
   Is CN=mycloud.quovadisglobal.com, OU=Cloud Services, O=QuoVadis Limited,
   L=Hamilton, ST=Pembroke, C=BM correct? [no]:yes
   ```

Note: QuoVadis is used for this example, The information that this summary displays should use your company's information.

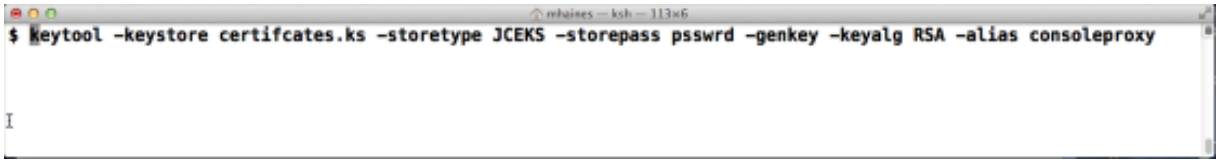
6. Confirm that you have access to this `keystore` file by entering a password. This uses "psswrd" as an example.
7. Type the key password for <http> psswrd (press Enter if it is the same as the keystore password).
8. Run the following command to obtain your CSR This creates the `http.csr` file.



2.2.2.3. Part II – Creating the CSR for the Proxy Service

To create the CSR for the proxy service

1. In the `keytool` directory, run the following command:



2. When prompted, type the FQDN (fully qualified domain name) to use for the Console Proxy Service certificate. Use the same FQDN as for the HTTP service certificate.
3. Type the following answers when prompted:

   What is your first and last name? [Unknown]: **mycloud.mydomain.com**

   What is the name of your organizational unit? [Unknown]: **MyCompanyDivision**

   What is the name of your organization? [Unknown]: **MyCompanyLegalName**

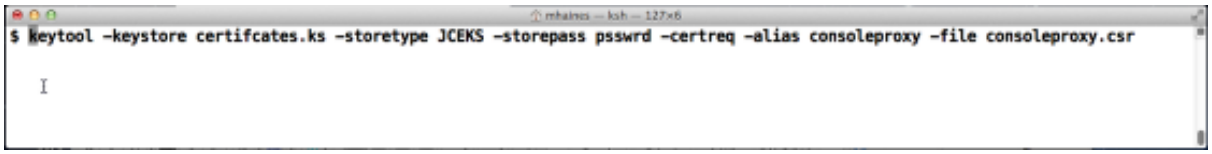
   What is the name of your City or Locality? [Unknown]: **CityOfMyCompany**

   What is the name of your State or Province? [Unknown]: **StateOfMyCompany**
4. Type **yes** to continue when `keytool` summarizes your entries.

   ```
   Is CN=mycloud.quovadisglobal.com, OU=Cloud Services, O=QuoVadis Limited,
   L=Hamilton, ST=Pembroke, C=BM correct? [no]:yes
   ```

   Note: QuoVadis, an international Certification Service Provider (CSP), has been used in the summary to provide a clear example. The information that this summary displays should use your company's information.
5. Confirm that you have access to this `keystore` file by entering a password. This uses "psswrd" as an example.
6. Type the key password for <http> psswrd (press Enter if it is the same as keystore password).

7. Next, run the following command to obtain your CSR (Certificate Signing Request). This creates the `consoleproxy.csr` file.



2.2.2.4. CSR Submission and Certificate Collection from QuoVadis

At this point you should have two separate CSRs, one for the HTTP service and one for the Console Proxy service (named `http.csr` and `consoleproxy.csr` in this example).

To complete the SSL Certificate Request forms and get access to the QuoVadis Trust/Link system

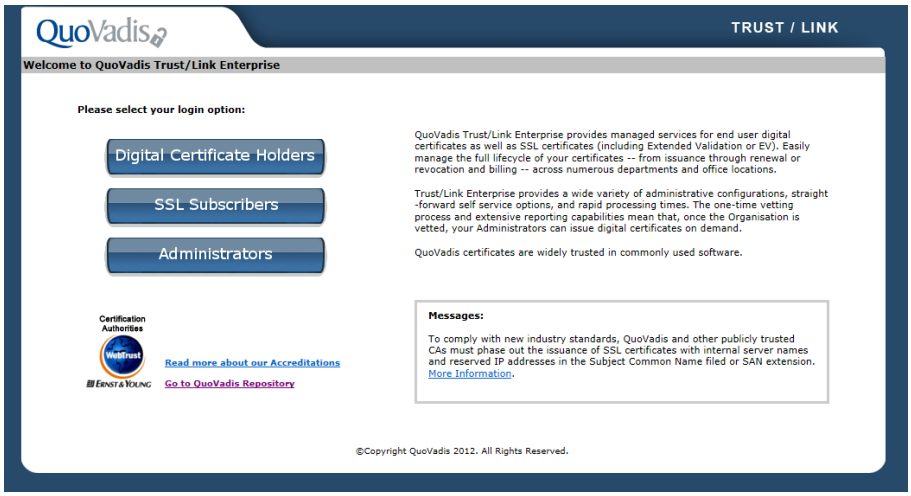
1. Copy both of these files to a computer that has Internet access.
2. Complete the QuoVadis SSL Certificate Request Forms to validate each SSL certificate request. The forms are available from: http://www.quovadisglobal.bm/sitecore/content/Bermuda/Manage/ApplicationForms.aspx
3. Submit the form to QuoVadis.

QuoVadis then validates your organization. After successful completion, you receive a login to the QuoVadis Trust/Link system.

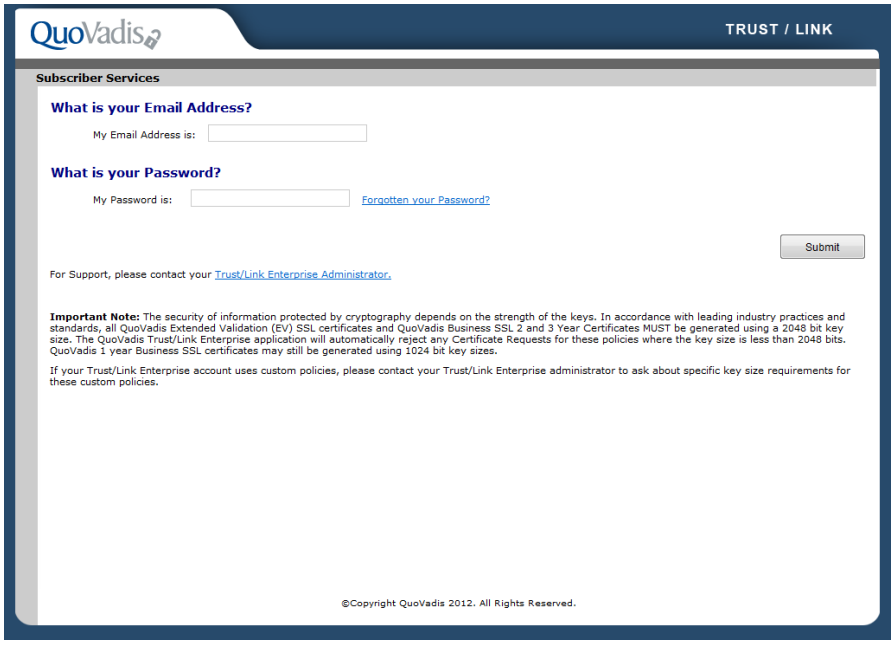
2.2.2.5. Part I – Submitting your CSRs

To submit your CSRs

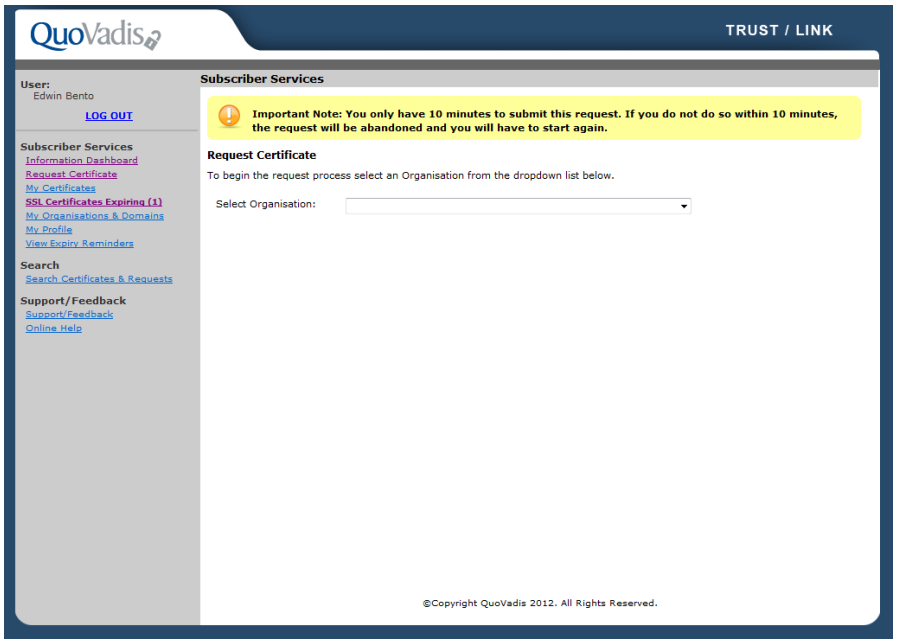
1. Go to https://tl.quovadisglobal.com.
2. Click **SSL Subscribers**.



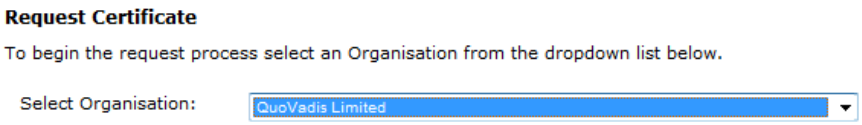
3. Log in with your email address and password.



4. Click the **Request Certificate** link under the **Subscriber Services** heading in the left pane. You have only 10 minutes to complete each request.



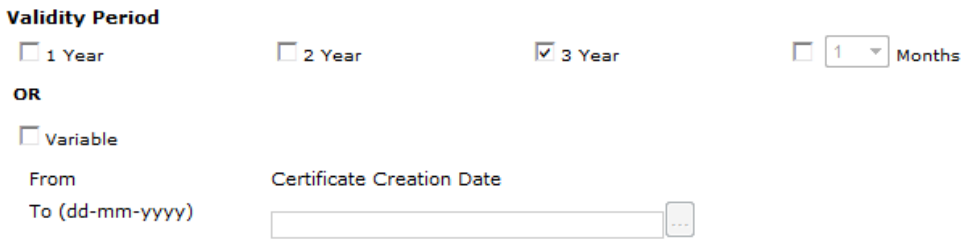
5. Select from the drop-down menu the approved organization to which you want to submit an SSL certificate.



6. Select the **Policy Template** that you want to use from the drop-down menu.

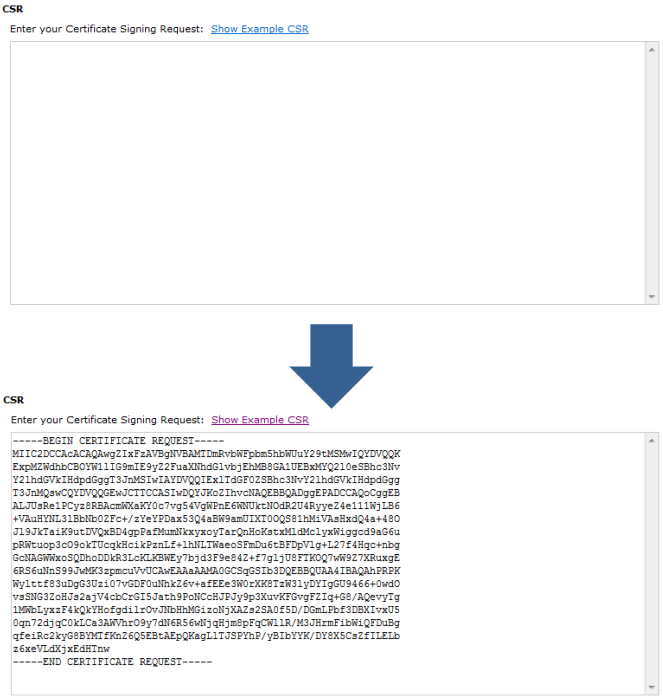
   If you do not have any policy templates to choose from, contact QuoVadis Support.

7. Select the **Validity Period** of the certificate.



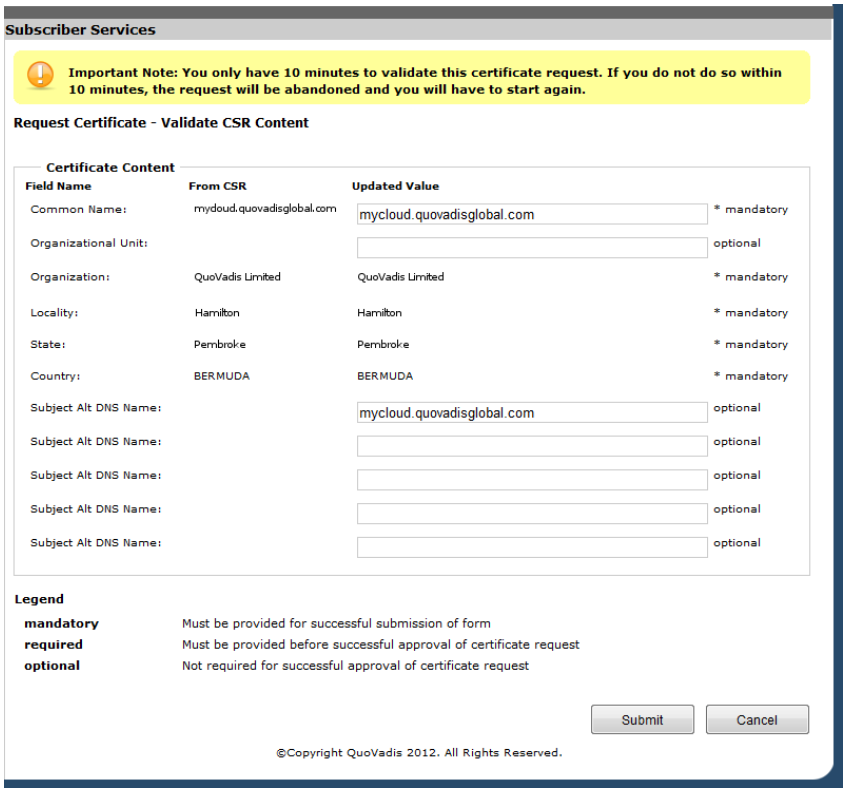
8. Optionally, select the Server Platform from the drop-down menu.

9. Open the HTTP CSR (or the Console Proxy CSR if this is your second run through) using a text editor such as VI on Linux or Notepad on Windows. Copy the contents from the text editor into your clipboard).
10. In Trust/Link, paste the contents of your CSR in the **Enter your Certificate Signing Request** field, and enter in all of the contents of your CSR, including the **BEGIN** and **END** lines.



11. Click **Submit**. The CSR that you submit is decoded and displayed in the Validate CSR Content screen.

12. Verify the content of the CSR and make changes, if necessary.



13. If the certificate requires any SAN fields, enter them into the **Subject Alt DNS Name** fields in the Certificate Content section. If any SAN fields are added, verify that the **Common Name** is included as the first SAN field.
14. When you are finished, click **Submit**. QuoVadis reviews the details of your certificate and contacts you if anything is incorrect. Your certificate is then approved.
15. Repeat this procedure for `consoleproxy.csr`

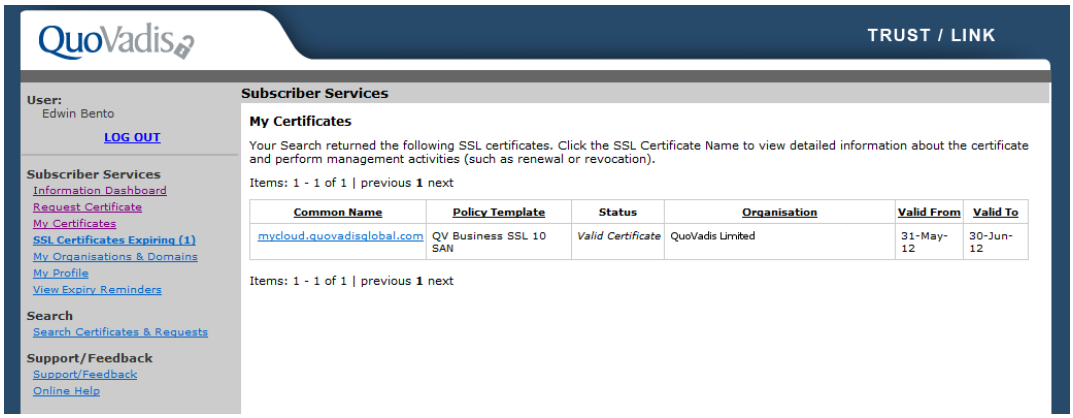
2.2.2.6. Part II – Obtaining your SSL Certificates

After the request has been approved, you receive an email informing you that your certificate is ready to download.

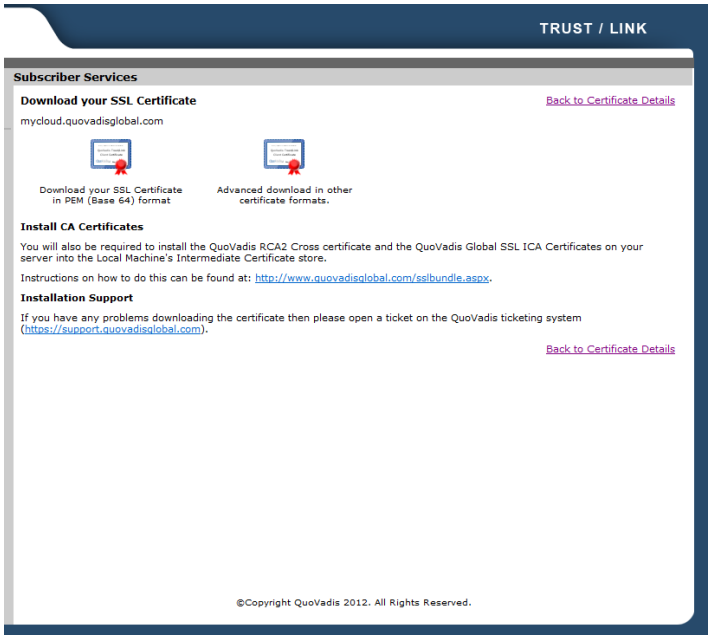
To obtain your SSL certificates

1. Go to https://tl.quovadisglobal.com.
2. Click **SSL Subscribers**.
3. Login with your email and password.
4. Click the **My Certificates** link under **Subscriber Services** in the left pane.

5. Click the **Common Name** of the certificate that you applied for. The status of the certificate should be **Valid Certificate**.



6. A summary of the certificate details is displayed. Click **Download**.
7. On the **Download your SSL Certificate** page, click the **Download your SSL Certificate in PEM (Base 64) format** icon. Rename this file to `http.crt` (assuming that you are downloading the certificate for the HTTP service).



8. Repeat this procedure for the Console Proxy Service SSL certificate. When you obtain this file, rename this file as `consoleproxy.crt`.

2.2.2.7. Installing your SSL Certificates

At this point you should have both SSL certificates for mycloud.mycompany.com, one for the HTTP Service (`http.crt`) and one for the Console Proxy Service (`consoleproxy.crt`).

To install your SSL certificates

1. Transfer the `http.crt and consoleproxy.crt` files to the `keytool` folder.
2. Download the QuoVadis Root CA 2 from: https://www.quovadisglobal.com/en-GB/QVRepository/~/media/Files/Roots/quovadis_rca2_der.ashx
3. Download the QuoVadis Global SSL ICA: https://www.quovadisglobal.com/en-GB/QVRepository/~/media/Files/Roots/quovadis_globalssl_der.ashx
4. Transfer both of these files to the `keytool` folder on the vCloud Director cell.

   The following files should be in the `keytool` folder:

   - `certificates.ks`
   - `http.crt`
   - `consoleproxy.crt`
   - `quovadis_rca2_der.crt`
   - `quovadis_globalssl_der.crt`
5. Run the following command to install the QuoVadis Root CA 2 certificate into the `keystore` file.

6. Run the following command to install the QuoVadis Global SSL ICA certificate into the `keystore` file:

7. Run the next two commands to install both the HTTP Services and Console Proxy Service certificates into the `keystore` file:

8. Run the following command to verify that all the certificates are imported correctly:

9. Move the `certificates.ks` file to a directory of your choosing. The directory used in this example is `/opt/keystore/`.
10. Remove or delete the .crt files from the vCloud Director cell after you have imported the certificates into the keystore.
    - `http.crt`
    - `consoleproxy.crt`
    - `quovadis_rca2_der.crt`
    - `quovadis_globalssl_der.crt`
11. Run the configuration script to configure vCloud Director. This script is located in the `/opt/vmware/vcloud-director/bin/configure` directory.
12. Specify the IP addresses for both the HTTP and Console Proxy Service.
13. When requested, enter the path to the `keystore` file. This is the folder where you uploaded your `certificates.ks` file.
14. Enter the path to the Java keystore that contains your SSL certificates and private keys:`/opt/keystore/certificate.ks`.
15. Type the password to access the keystore file. Also type private key passwords for the certificates.

    Please enter the password for the keystore: **psswrd**

    Please enter the private key password for the 'http' SSL certificate:

    Please enter the private key password for the consoleproxy SSL certificate:

16. Finish the configuration according to your setup. When the process is complete, your SSL certificate should work with vCloud Director.

2.2.3 Design Implications

- When using SSL certificates it is important to understand and evaluate what the different types of SSL certificates are available to you for your specific requirement.
- In a production environment, do not configure vCloud Director to use self-signed certificates. This is a very poor security practice. Self-signed certificates are certificates that are digitally signed by the private key corresponding to the public key included in the certificate. This is done in place of a CA signing the certificate. By self-signing a certificate, you are attesting that you are who you say you are. No trusted third party is involved to verify the identity of the system that owns the certificate.
- Self-signed certificates do not have a valid chain of signatures leading to a trusted root certificate. They provide a weaker form of security because, while you can verify such a certificate is internally consistent, anyone can create one, so by examining the certificate, you cannot know if it is safe to trust the issuer or the site the certificate is coming from. Nevertheless, self-signed certificates are common. For example, vCenter installations use a self-signed certificate by default.
- The server key store should be considered highly sensitive since a compromise of the server key allows impersonation of the server and/or access to the encrypted traffic. Java keystores provide a method of securely storing private keys and their associated certificates, protected by a password. vCloud Director supports only the JCEKS format for keystore files. (Other formats that Java supports include PKCS12 and JKS. JKS is less secure, so it is not recommended).

3. Organization Virtual Datacenter Examples

Understanding how to configure and setup the various allocation models is key to building out your vCloud. Allocation models define the way resources are allocated from the provider virtual datacenter to a vCloud Director organization virtual datacenter. They also define the way resources can be used when deploying vApps within the vCloud Director organization's virtual datacenter. The allocation models are:

- Pay-As-You-Go.
- Reservation Pool.
- Allocation Pool.

Because allocation models are specific to a particular organization, they should be treated individually for each consumer. Although many providers use a starting template approach, allocation models are specific to a particular organization and they should be configured to meet the needs of the consumer that will be using them. The provider must understand the individual consumer's requirements in the context of the available configuration options.

3.1 Pay-As-You-Go Allocation Model

Deployment Models: private, public.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

3.1.1 Background

The following is an example of a vCloud provider setting up the Pay-As-You-Go model based on consumer's criteria.

3.1.2 Use Case

A new customer known as Company2 is requesting resources on Company1's vCloud and has asked the vCloud provider to obtain capacity in the Company1 vCloud. Company2 wants a model that provides the following:

- They want to pay for resources as they use them, and have unlimited access to resources within their organization virtual datacenter.
- Very high performing virtual machines.
- Highest level of memory guarantees. Memory performance is a greater concern than CPU.
- Virtual machines with multiple vCPUs that have a minimum speed of at least 2GHz, and some level of CPU guarantee.

The Pay-As-You-Go model was chosen because it satisfies Company2's requirements and applies the resources the consumer wants the same way to each virtual machine they deploy.

3.1.2.1. Assumptions

- Company1 has a single configured provider virtual datacenter.
- Networking is routed – vCloud Networking and Security Gateways are deployed and configured after the organization is configured.

3.1.2.2. Organization Functional Requirements:

- 100% memory reservation.
- Agreed to 50% CPU reservation.
- Minimum CPU speed 2GHz.
- Unlimited resources within the organization virtual datacenter.
  - No CPU or memory quota.
  - No maximum number of virtual machines.
- There are no specific storage requirements.
  - Storage is thin provisioned.
  - Fast provisioning is disabled.

Based on the requirements, Company1 will use the following settings to configure Company2's organization.

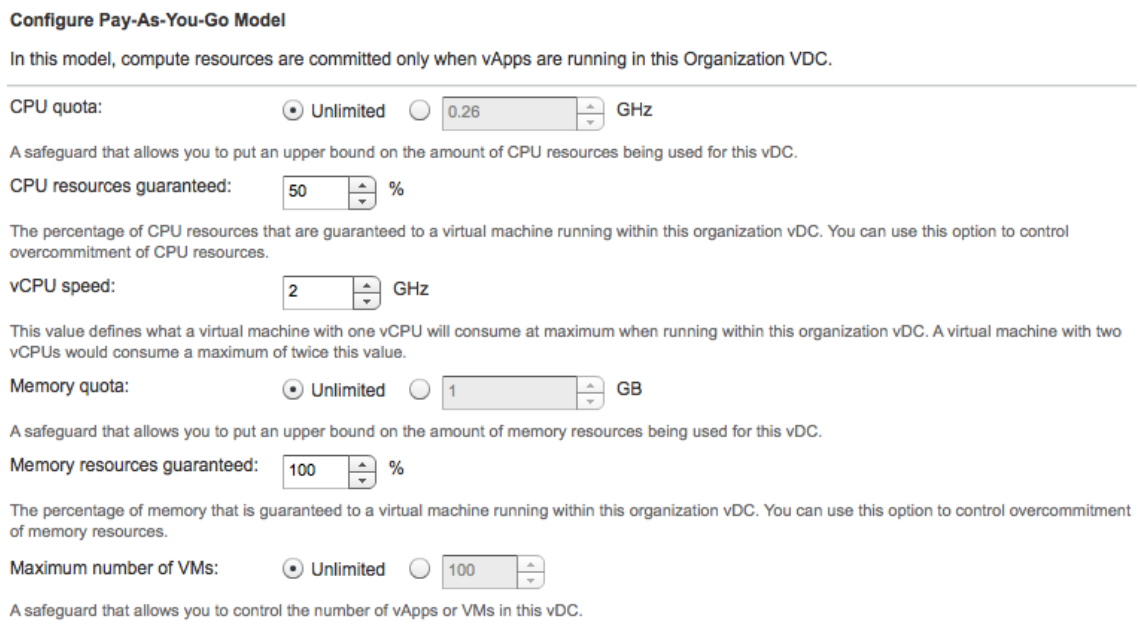
Table 3. Company2 Pay-As-You-Go Organization Settings

Setting	Memory Reservation
CPU quota	Unlimited
CPU resources guaranteed	50%
vCPU speed	2GHz
Memory quota	Unlimited
Memory resources guaranteed	100%
Maximum number of VMs	Unlimited

3.1.3 Example

The following figure shows the vCloud Director configuration for Company2's organization.

Figure 4. Pay-As-You-Go Settings



Values for settings correspond to Company2's requirements.

- **CPU quota** – Provisioning stops when the virtual datacenter has reached the configured amount. For example, if this is set to 100GHz and all virtual machines are provisioned with 1GHz CPUs, when 100GHz worth of CPU is deployed no more will be provisioned. This can be a combined total based on the number of vCPUs on each virtual machine.
- **CPU resources guaranteed** – This sets a per virtual machine reservation on CPU based on a given percentage.
- **vCPU speed** – This sets a per virtual machine CPU limit to the specified amount.
- **Memory quota** – This works the same for memory as above with CPU. Setting an amount prevents provisioning of more virtual machines when that number is reached.
- **Memory resources guaranteed** – This sets a per virtual machine reservation on memory based on a given percentage.
- **Maximum number of VMs** – This is a hard limit on the organization virtual datacenter for the total number of virtual machines that can be deployed. This can be useful to prevent overcommitment.

Note: The vSphere resource pool expandable reservation for both CPU and memory should be **Enabled**.

These settings can be changed to meet the consumer's functional requirements for performance and cost over time.

3.1.4 Design Implications

Because this allocation model assigns all settings on a per virtual machine basis, any updates to this model requires a shutdown and restart of the virtual machines. Based on the selected settings, this could be considered one of the best performing models in terms of guaranteeing the allocated virtual machine resources. This is because each virtual machine is always guaranteed its settings and can go to the root resource pool if needed. Company2 will get 100% of physical memory to all virtual machines, and 50% physical CPU.

Essentially, this model provides unlimited resources within the organization virtual datacenter, so providers must be diligent in capacity planning. Company1 must proactively monitor the provider virtual datacenter for resource availability to make sure that new virtual machines can continue to be deployed and powered on.

The Pay-As-You-Go model also has the capability to leverage elastic provider virtual datacenters. The Pay-As-You-Go model can use this added capacity automatically, and is transparent to the consumer. They can continue to deploy virtual machines as long as there is capacity to meet their requirements.

3.2 Reservation Pool Model

Deployment Models: private, public.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

3.2.1 Background

The following is an example of a provider creating the Reservation Pool model based on the consumer's criteria.

3.2.2 Use Case

A new customer known as Company2 is requesting resources on Company1's vCloud and has asked the vCloud provider to obtain capacity in the Company1 vCloud. Company2 wants a model that provides the following:

- A defined amount of memory and CPU resources that will be consistent for billing. They have not yet determined the number of virtual machines they will have, but they want consistent billing.
- A dedicated pool of resources to distribute to virtual machines as they see fit when deploying workloads. Based on the initial pool of resources they request, Company1 will also apply a total virtual machine limit to the pool to help Company2 limit overcommitment.
- No fast provisioning, but thin provisioning is acceptable.

Settings can be adjusted as needed to meet the needs of Company2's vApps.

Based on these requirements, the Reservation Pool model was chosen as the best fit to meet Company2's needs. This model gives them dedicated resources to start with, and they have control over resource allocation to the virtual machines from vCloud Director.

3.2.2.1. Assumptions

- Company1 has a single configured provider virtual datacenter. Storage performance is predetermined.
- Networking will be routed – vCloud Networking and Security gateways will be deployed and configured after the organization is configured.
- Customer will control and maintain the individual virtual machine resource configurations.

3.2.2.2. Organization Functional Requirements

- 25GB of memory.
- 25GHz of CPU.
- 25 maximum virtual machines – Determined by Company1 based on the organization sizing to prevent overcommitment.
- Personal control over the individual virtual machine resource configurations.
- No fast provisioning but thin provisioning enabled.

Based on the requirements, Company1 will use the following settings to configure the Company2's organization.

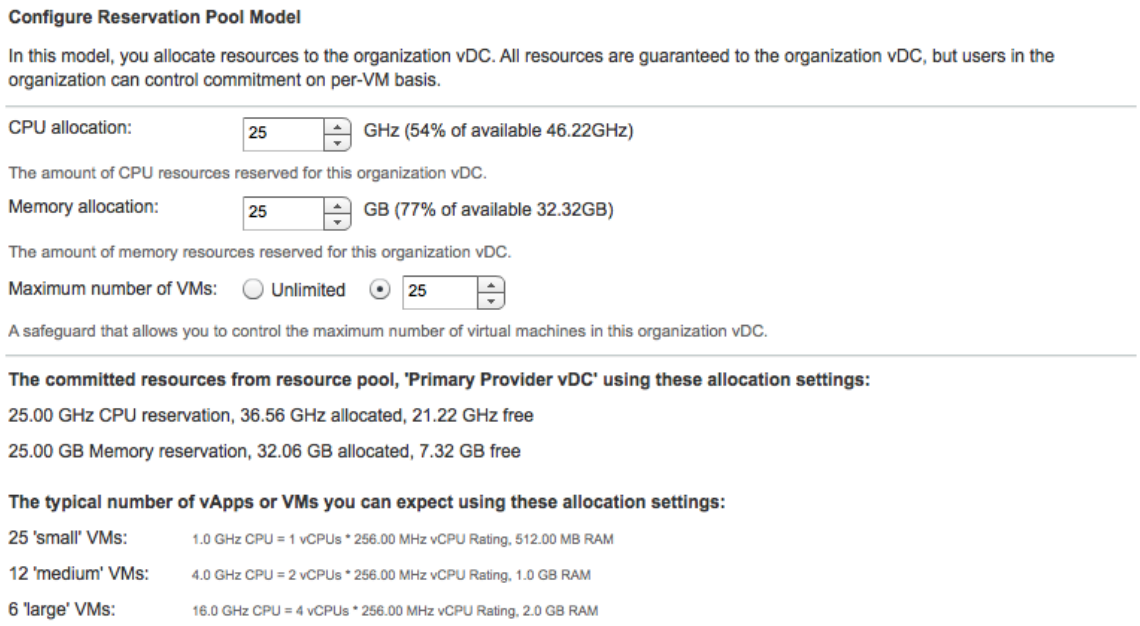
Table 4. Company2 Reservation Pool Organization Settings

Parameter	Setting
CPU allocation	25GHz
Memory allocation	25GB
Maximum number of virtual machines	25

3.2.3 Example

The following figure shows the vCloud Director configuration for Company2's organization.

Figure 5. Reservation Pool Settings



The values for settings correspond to Company2's requirements.

- **CPU allocation** – This sets the CPU on the vSphere resource pool with a reservation and limit equal to the selected amount.
- **Memory allocation** – This sets the memory on the vSphere resource pool with a reservation and limit equal to the selected amount.
- **Maximum number of VMs –** This is a hard limit on the virtual datacenter for the total number of virtual machines that can be deployed. This can be useful to prevent overcommitment.

Note: The vSphere resource pool expandable reservation for both CPU and memory should be **Disabled**.

These settings can be changed to meet the consumer's functional requirements for performance and cost over time. In fact, changes to these settings can be made on the fly to increase the amount of resources. Depending on the number of vApps deployed, carefully consider the potential impact before reducing values.

3.2.4 Design Implications

Because this allocation model assigns all setting on a resource pool basis, updates to this model do not require a shutdown and restart of the virtual machines to pick up the changes. Based on the settings chosen, this model could be considered as one of the best models because the consumer has control over how the pool of resources is divided among the virtual machines.

The provider must monitor the use of the **Maximum number of VMs** setting. If this is left as unlimited and the consumer does not set any per virtual machine reservations, there is potential for overcommitment. A consumer can, in theory, deploy more or larger virtual machines than the pool is configured for (additional resources such as memory could be paged out to satisfy the configuration). Assigning a limit to the maximum total virtual machines can help control this and keep the pool within the size it should be.

Huge virtual machines can be created using this model, but it is important to understand the role that overhead reservations play along with the standard per virtual machine reservations. Understanding the effect of these settings on vSphere is crucial for implementation.

See the following external reference for more information: *VMware vSphere 5.1 Clustering Deepdive* (Chapter 13). Available from Amazon.com.

3.3 Allocation Pool Model

Deployment Models: private, public.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

3.3.1 Background

The following is an example of a vCloud provider setting up the Allocation Pool model based on consumer's criteria.

3.3.2 Use Case

A new customer known as Company2 is requesting resources on Company1's vCloud and has asked the vCloud provider to obtain capacity in the Company1 vCloud. Company2 wants a model that provides the following:

- A model that combines the characteristics of the Pay-As-You-Go per virtual machine settings with the pool-based aspect of the reservation pool.
- High performing virtual machines, but they do not want the management overhead of setting resource options for each virtual machine. They want to start with a set amount of resources, with a portion automatically guaranteed. They are not sure how many virtual machines they will need to deploy.
- No fast provisioning, but thin provisioning is acceptable.

Based on the initial pool of resources Company2 requested, Company1 will apply a total virtual machine limit to the pool to help Company2 limit overcommitment. Company1 can adjust these settings as needed to meet more or less demand from Customer2's vApps, but some virtual machine restarts would be needed.

Based on these requirements, the Allocation Pool model was chosen as the best fit to meet Company2's needs. This model gives them dedicated resources to start with, and preset guarantees are applied to each virtual machine along with the total guarantee set in the resource pool.

3.3.2.1. Assumptions

- Company1 has a single provider virtual datacenter configured. Storage performance is predetermined.
- Networking will be routed – vCloud Networking and Security Gateways will be deployed by vCloud Director at network creation time after the organization is configured.

3.3.2.2. Organization Functional Requirements

- 25GB of memory with 100% guarantee – This is applied to the pool and the individual virtual machines automatically.
- 25GHz of CPU with 50% guarantee – Applied to the pool, but not to the virtual machines.
- 12 maximum virtual machines – Determined by Company1 based on the organization sizing to prevent overcommitment.
- vCPU speed of at least 2GHz.
- No fast provisioning.

Based on the requirements, Company1 will use the following settings to configure Company2's organization.

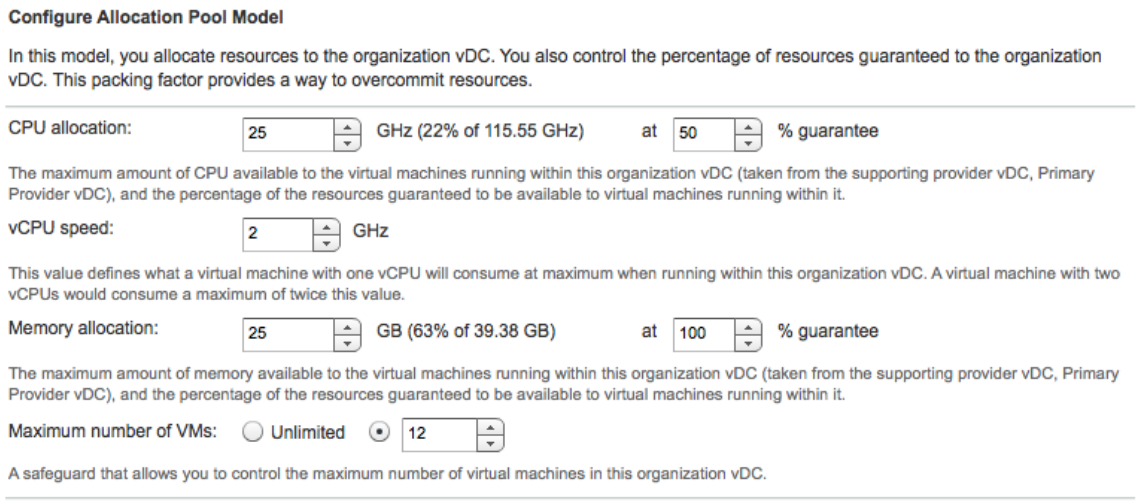
Table 5. Company2 Allocation Pool Organization Settings

Parameter	Setting
CPU allocation	25GHz
Memory allocation	25GB
vCPU speed	2GHz
Maximum number of virtual machines	12

3.3.3 Example

The following figure shows the vCloud Director configuration for Company2's organization.

Figure 6. Configure Allocation Pool Screen



Values for settings correspond to Company2's requirements.

- **CPU allocation** – This sets the CPU on the vSphere resource pool with a reservation and limit equal to the to the selected amount.
  - **% guarantee –** This is the amount of the vSphere reservation on the resource pool.
- **vCPU speed –** This sets a per virtual machine CPU limit to the specified amount .
- **Memory allocation –** This sets the memory on the vSphere resource pool with a reservation and limit equal to the specified amount.
  - **% guarantee –** This is the amount of the vSphere reservation on the resource pool and on the individual virtual machines.
- **Maximum number of VMs –** This is a hard limit on the virtual datacenter for the total number of virtual machines that can be deployed. This can be useful to prevent overcommitment.

Note: The vSphere resource pool that corresponds to the virtual datacenter has an expandable reservation on both CPU and memory **Disabled**.

These settings can be changed to meet the consumer's functional requirements for performance and cost over time. In fact, changes to these settings can be made on the fly to increase the amount of resources. Depending on the number of vApps deployed, carefully consider the potential impact before reducing values.

3.3.4 Design Implications

Because this model assigns settings to both a resource pool and the virtual machines, any updates require a shutdown and restart of the virtual machines. The percentage guarantee for memory is applied to every virtual machine in this virtual datacenter, similar to the Pay-As-You-Go model. Additionally, the settings are applied to the resource pool in vSphere like the Reservation Pool model.

With the settings shown in Figure 6, 100% memory reservation is available on the vSphere resource pool, and 50% of the CPU reservation is available. Each virtual machine created also requires a 100% virtual machine memory reservation within the pool to run. After all of the pool's resources are assigned, no more virtual machines are allowed to run, but the settings can be adjusted. Because the resource pool expandable reservation is disabled, in most cases VMware admission control will prevent too many virtual machines from being powered on.

Lowering the percentage guarantee amount below 75% or even 50% can also create challenges as the resource pool gets only a fraction of the configured resources shown as **Available Reservations**. Virtual machines are still required to reserve those same values at power on. When using the Allocation Pool model, using higher values is always better to prevent any limitations within the organization virtual datacenter.

The Allocation Pool model can also leverage elastic provider virtual datacenters in vCloud Director. The Allocation Pool model can use this added capacity automatically, without the affecting Company2. They can continue to deploy virtual machines as long as there is capacity to meet their requirements.

Understanding the effect of these settings on vSphere is crucial to implementation.

As with all of the allocation models, the provider needs to monitor the customer's virtual datacenters and keep the customer informed about resource usage. If virtual datacenter runs out of resources, vCloud Director sends an error message to the customer.

See the following for more information: *VMware vSphere 5.1 Clustering Deepdive* (Chapter 13) available from Amazon.com.

3.4 Service Provider Performance Offerings

Deployment Models: private, public.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

3.4.1 Background

During any provider and consumer conversation the topic of virtual machine performance always arises. In some cases this is referred to as one element of a Service Level Agreement (SLA). In this document, the aspect of a tenant's performance is examined in relation to the three allocation models.

Note: Higher performance levels are defined here as guaranteeing physical resources from the provider virtual datacenter to the consumer. Actual virtual machine performance might vary based upon the application running within the virtual machine. However, guaranteeing required resource availability to a virtual machine provides the best performance.

We have already discussed examples of how to create virtual datacenters based on various use cases. The intent here is to expand on the tenant requirements that may drive decisions to the different allocation models. Thinking about allocation models in the context of performance rather than uptime or availability is just another option.

3.4.2 Use Case

Company1 has built a new vCloud Director environment and is working on how to present options to customers for performance-based virtual datacenters. Knowing that the vCloud Director allocation models are largely reservation based, they want to determine the best set of requirements and considerations to take into account when talking to customers. They want to tailor offerings to the customer's needs, but have not finalized the service offerings.

3.4.2.1. Assumptions

- The same provider virtual datacenter is backing the various allocation models.
- Storage is currently the same within the provider virtual datacenter.
- The catalog of virtual machines will vary in size.
- An elastic virtual datacenter will be used, if needed (assuming it can be used).

To develop a menu of options for customers three levels of performance have been determined for each allocation model. These can be modified as customer needs change. Company1 defines performance categories as:

- Low
- Medium
- High

This is not the same as how virtual machine templates are defined (Small, Medium, and Large). The goal is to balance the machine configurations with the actual allocation model under which customers are running.

3.4.3 Example

Company1 originally set up template-based organization virtual datacenters that would all provide the same consumer offering within each available model as shown in the following table.

Table 6. Company1 Pay-As-You-Go Offering

Parameter	Low	Medium	High
vCPU speed	1GHz	2GHz	2.5GHz
CPU guarantee	0%	25%	50%
Memory guarantee	0%	50%	100%
Memory quota	N/A	N/A	N/A
CPU quota	N/A	N/A	N/A
Maximum number of virtual machines	Unlimited	Unlimited	Unlimited

Pay-As-You-Go is an “all you can eat” model with the only restriction being the amount of resources available in the provider virtual datacenter. The settings above are assigned individually on a per virtual machine basis. In effect, the memory and CPU guarantees allocate physical resources to these virtual machines. Therefore, a 100% guarantee means that each virtual machine is granted 100% of its requested resources within the virtual datacenter, as is represented in the case of memory for High as long as the physical resources are available. In this example, a 2.5GHz CPU with 50% CPU and 100% memory guarantee could effectively be the highest performing virtual machine possible.

Table 7. Company1 Reservation Pool Offering

Parameter	Low	Medium	High
CPU allocation	15GHz	25GHz	50GHz
Memory allocation	15GB	25GB	50GB
Maximum number of virtual machines	15	25	50

With the reservation pool model, the responsibility is on the consumer to assign physical resources from the pool to the virtual machines. This model effectively reserves 100% of the physical resources configured for CPU and memory from Company1's provider virtual datacenter. CPU and memory resources cannot be used by other customers. Consumers do not necessarily get better performance, but there is the potential for higher performing and lower performing virtual machines within the consumer's organization. The consumer must decide which virtual machines will be configured for higher or lower performance within the given pool of resources.Company1 has provided resources that the consumer can allocate however is desired. The levels represent the larger amounts of resources a customer has available. This increases the potential for more individual virtual machines to perform better, provided they are configured correctly.

Table 8. Company1 Allocation Pool Offering

Parameter	Low	Medium	High
CPU allocation	15GHz	25GHz	50GHz
CPU guarantee	50% (7.5GHz)	75% (18.75GHz)	100% (50GHz)
vCPU speed	1GHz	2GHz	2.5GHz
Memory allocation	15GB	25GB	50GB
Memory guarantee*	50% (7.5GB)	75%(18.75GB)	100% (50GB)
Maximum number of virtual machines	15	25	50

*Memory is assigned to an individual virtual machine based on the percentage.

With the Allocation Pool, the available resources in the pool are determined by the guarantees assigned. Listed above are the actual available resources within the pool for the consumer to use. This is similar to the Reservation Pool model. Effectively the high performance option is truly that, as it gives the pool a 100% guarantee not only on the resource pool, but also the individual virtual machines. The High option is almost a full combination of the other two models. The only real difference is when Low or Medium is used, some virtual machines can access the extra, unreserved resources depending on the configuration.

3.4.4 Design Implications

High performance relates to guaranteeing physical resources in the provider virtual datacenter. Providers must keep in mind that this could mean lower consolidation ratios on the provider clusters. In all cases Company1 has imposed a “total number of virtual machines” limit on the consumers. This is to mitigate some of the considerations below.

- Pay-As-You-Go – With this model the consumer may get performance that is easier to manage, but due to its “all you can eat” nature, Company1 must closely monitor the provider virtual datacenter. If the provider virtual datacenter is low on resources, it can affect the consumer’s ability to add more virtual machines.
- Reservation Pool – With this model it is up to the consumer to use resources responsibly to get the desired performance within the pool. Otherwise, higher consolidation ratios can be achieved, but performance can suffer. In addition, this can also affect other consumers on the same datastores due to swapping.
- Allocation Pool – The model can mitigate some of the issues found in the Reservation Pool model. Due to the fact this is a hybrid of the Pay-As-You-Go and Reservation Pool models, it handles everything provided you use the high performance configuration. By using the high performance configuration you get a dedicated pool that the consumer cannot grow out of (like the reservation pool). Each virtual machine in the pool gets 100% guarantee of the available resources. When the pool is completely consumed, provisioning stops.

Company1 has designed the deployment with some aspects of consumer performance in mind, not only with each allocation model, but also with levels within each model. These templates are not static—they are just an initial starting point. Customer requirements can drive customization and changes to these options. Company1 will bill customers for any changes required. They feel consumers can start with these models and then modify them as needed.

Note: It is not possible for a consumer to change between models without downtime and migration. Modifying settings of each allocation model is possible but some virtual machine reboots may be required.

See the following for more information: *VMware vSphere 5.1 Clustering Deepdive* (Chapter 13) available from Amazon.com.

4. Networking Examples

4.1 vApp Load Balancing with vCloud Networking and Security Edge

Deployment Models: private, public.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

4.1.1 Background

The vCloud Networking and Security Edge Gateway (Edge Gateway) introduced by vCloud Director supports and exposes a powerful load balancer to the organization administrator. In this document a simple example of its configuration is demonstrated.

4.1.2 Use Case

An organization administrator is setting up a simple n-tier application comprised of front end Web servers and back end database servers. For high availability and scaling, the administrator wants to load balance these Web servers. The administrator wants to avoid using in-guest techniques and does not want to deploy additional virtual machines to deliver this load balancing service because of the burden of having to manage additional infrastructure workloads. The administrator has decided to use the out-of-the-box load balancing service provided by the Edge Gateway as the easiest way to meet these needs. The administrator wants to use public IP addresses to be able to define a virtual IP that balances (round-robin) two Web servers that are instantiated in an organization.

4.1.3 Example

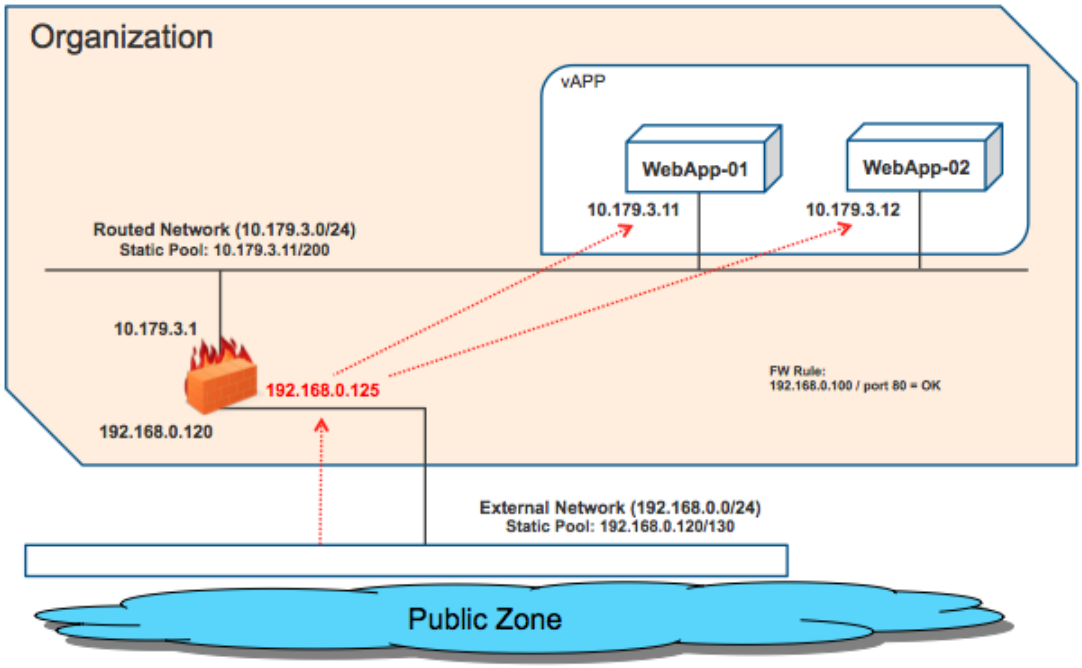
The Edge Gateway is connected to one side to the vCloud Director external network, which connects directly to the Internet. It is also connected to two routed organization virtual datacenter networks. Our focus is on the network named “Org-External-Routed-Internet” because this is where the Web servers to be load balanced are connected.

Table 9. Network Device Information

Device	Location	IP/Netmask	Notes
Internet-NAT	External network	192.168.0.0/24	Class C network
	Static IP pool	192.168.0.120–130	10 public IP addresses
Org-External-Routed-Internet	Organization virtual datacenter network	10.179.3.0/24	Class C network
	Static IP pool	10.179.3.11–200	190 internal IP addresses
Edge-Gateway	COE organization	External: 192.168.0.120	Internet-NAT
		Internal: 10.179.3.1	Org-External-Routed-Internet
WebApp-01	COE organization	10.179.3.11	Guest IP address
WebApp-02	COE organization	10.179.3.12	Guest IP address

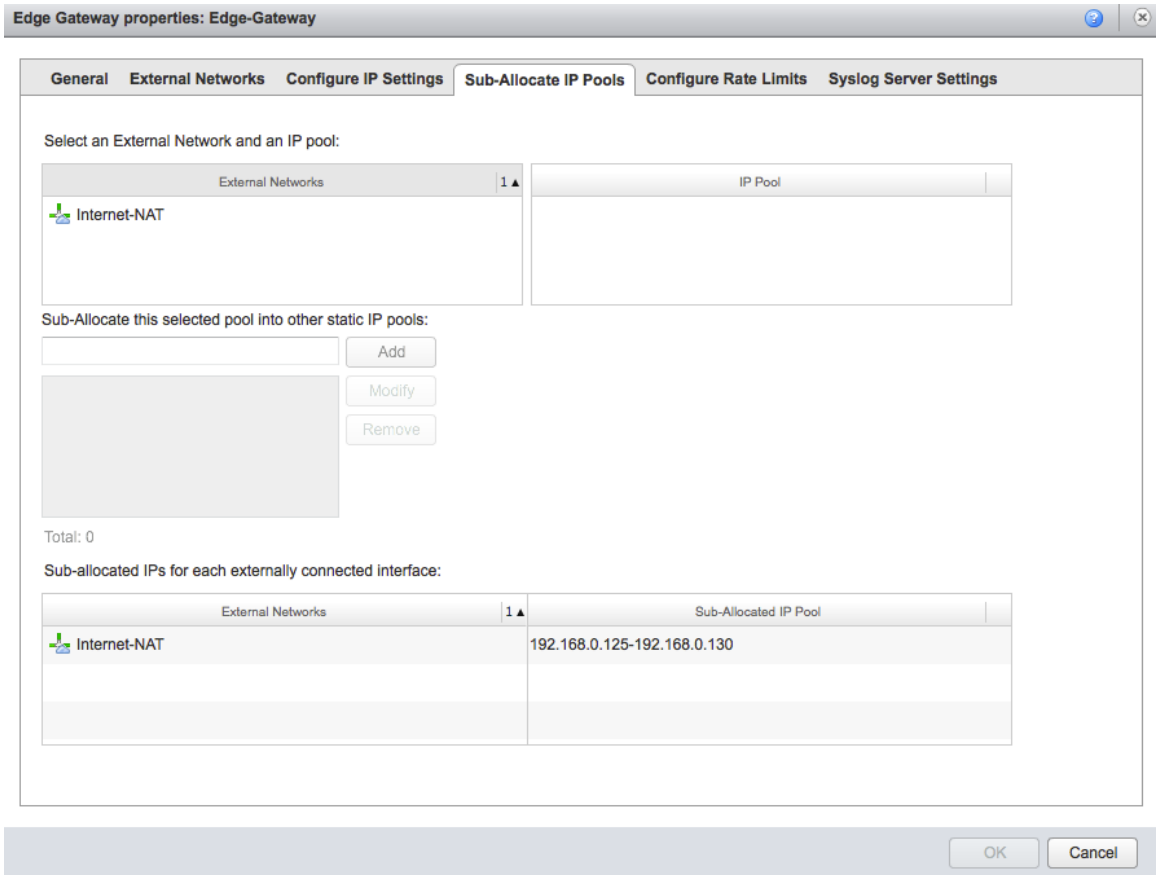
Figure 7 shows the components of a typical load balancing arrangement.

Figure 7. Graphical Summary of Components

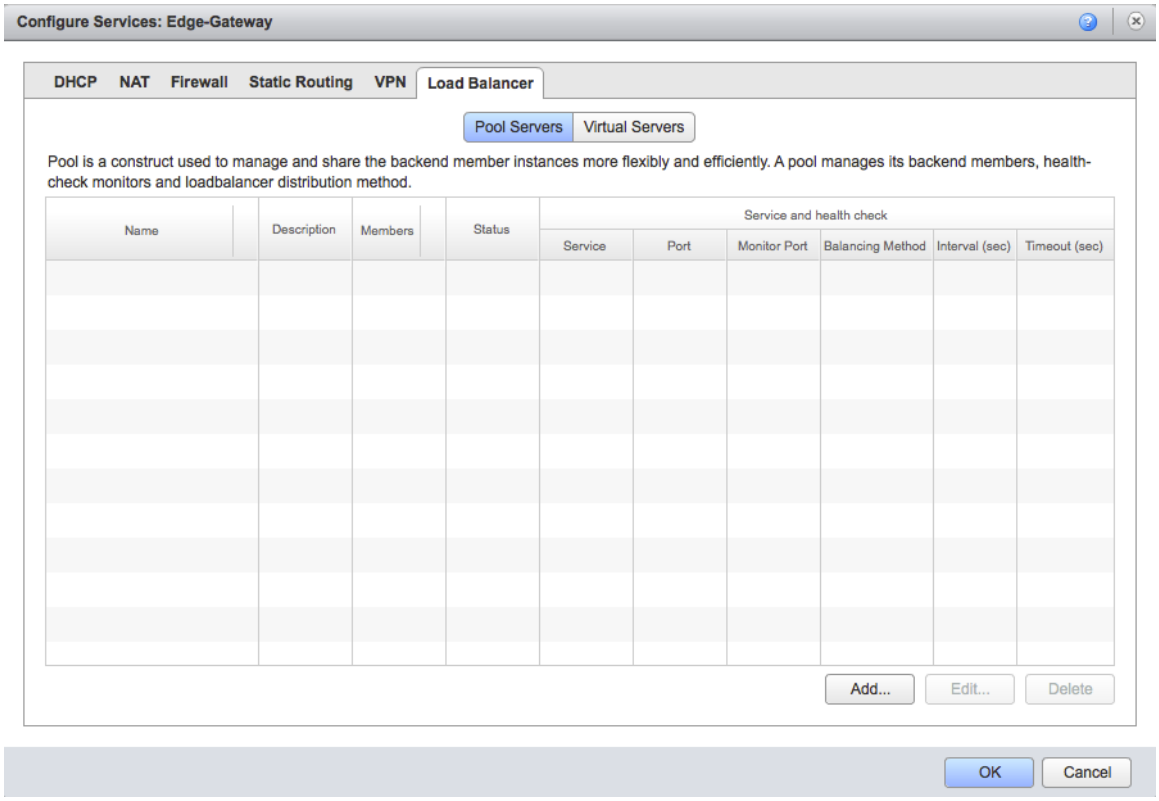


To create load balancing with vCloud Networking and Security Edge

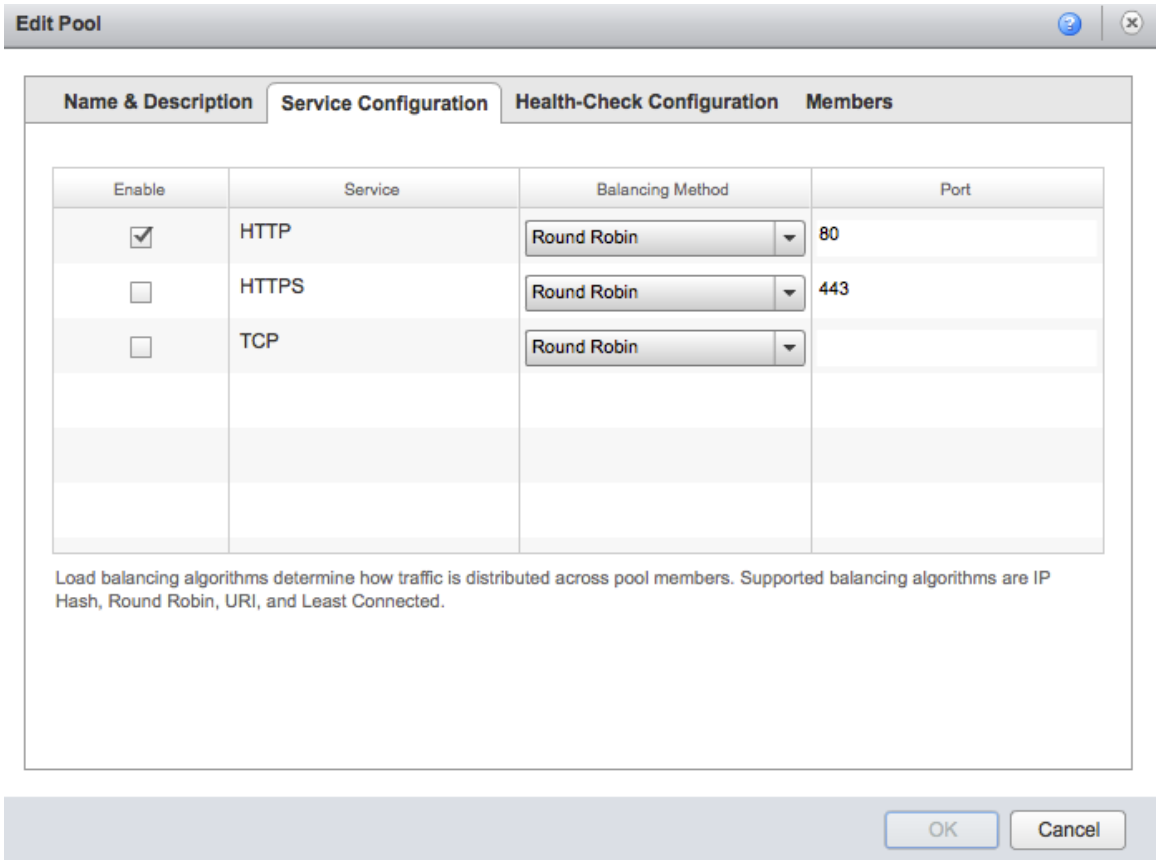
1. To create a load-balancing rule the organization administrator must use a public IP address from the external network that was preallocated by the vCloud administrator. In this example, the vCloud administrator reserved for this vCloud Networking and Security Edge instance the public IP address range 192.168.0.125–130 as indicated in the vCloud Networking and Security Edge Gateway properties. One of these reserved IP addresses will be used later as the virtual IP address.



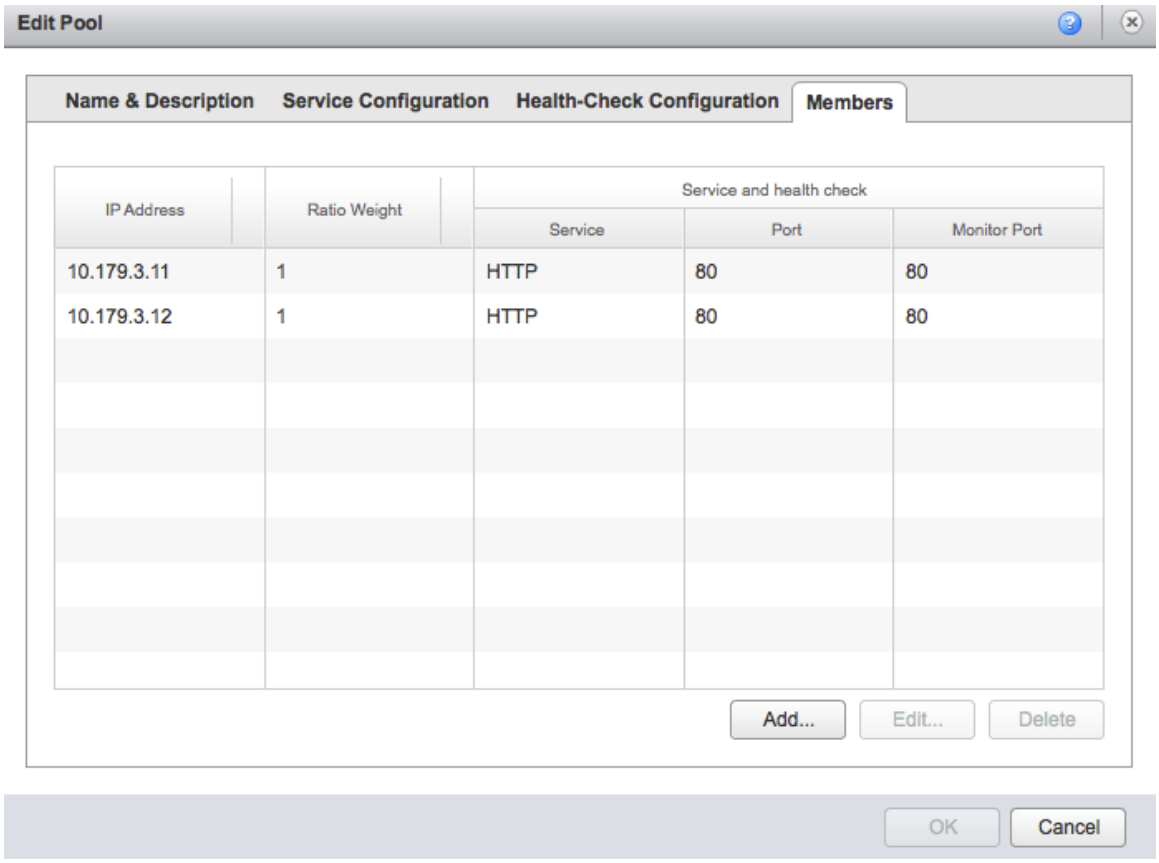
2. Open the vCloud Networking and Security Edge Gateway services page and click the **Load Balancer** tab to configure the load-balancing rule.



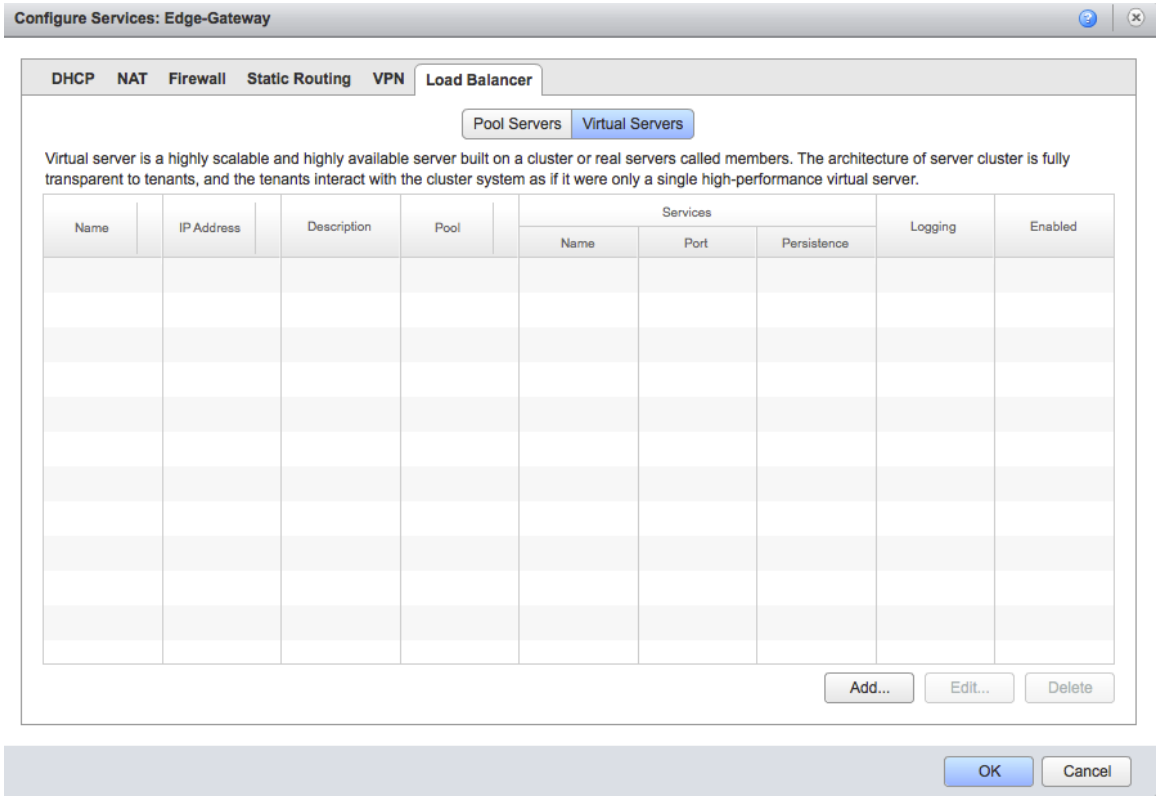
3. Click **Pool Servers**. List the Web servers that are to be load balanced.



4. Configure the two Web servers in this pool and use the vCloud Networking and Security Edge Gateway to balance HTTP traffic from the virtual server to the Web servers on destination port 80 in a round-robin.

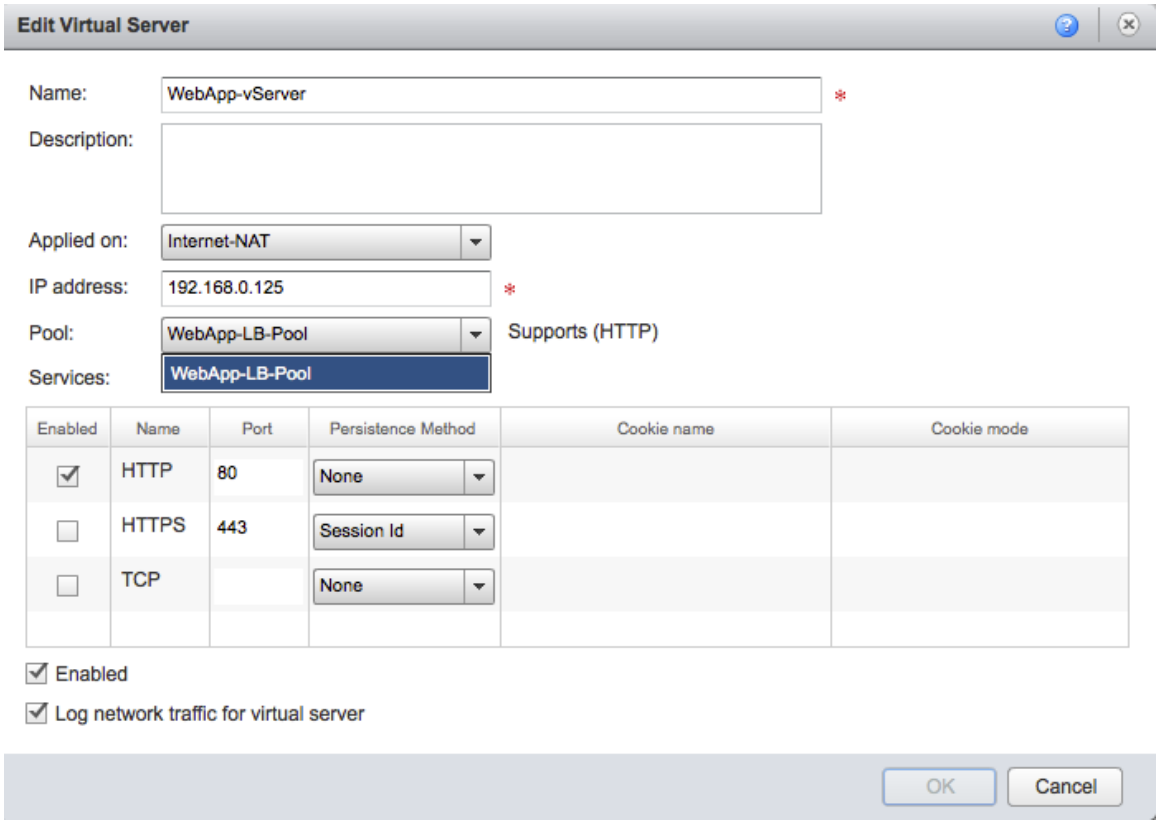


5. Now that a pool of servers is configured, a virtual server can be configured from the following view of the **Load Balancer** tab.



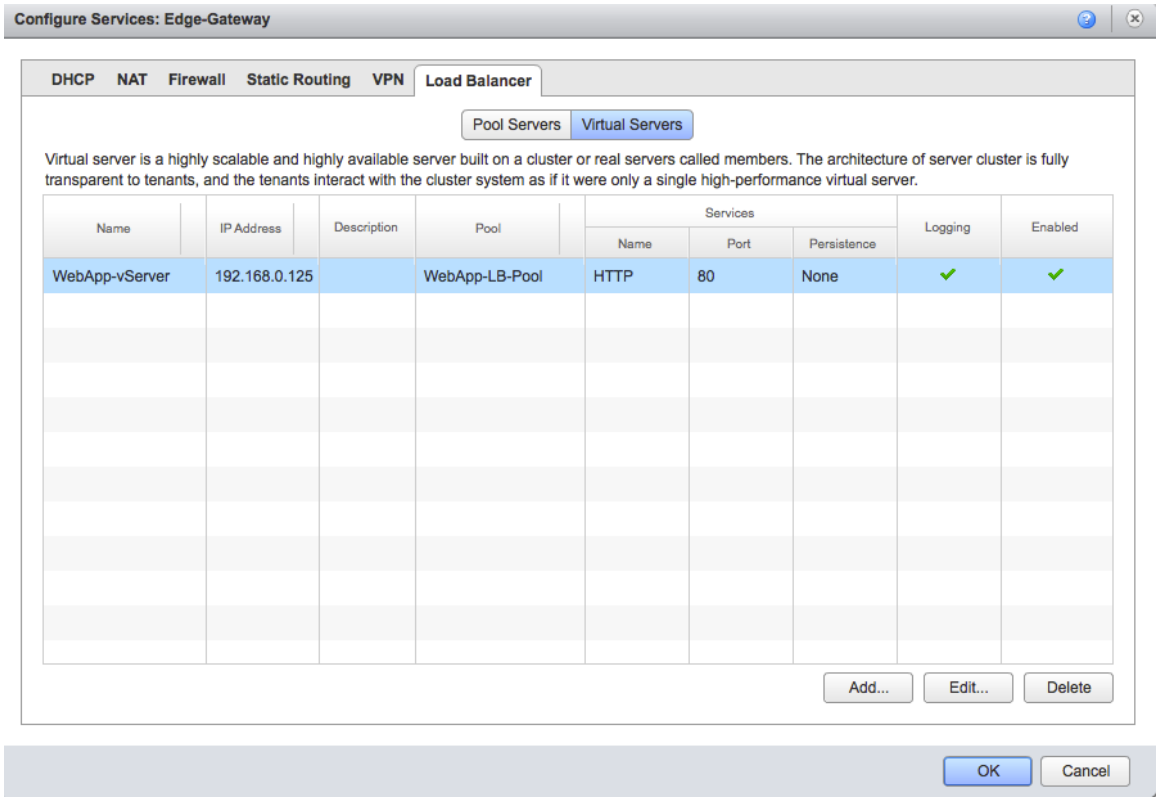
6. You can configure a new virtual server to use by selecting the previously defined pool and allowing the services you want (Figure 7). This is the pool we created in the previous step and that contains the two front end Web servers.

   When the configuration is finalized, this new virtual server is displayed in the **Load Balancer** page of the vCloud Networking and Security Edge Gateway services.



Note: The 192.168.0.125 IP address used in the example is one of the public IP addresses (on the external network) assigned to this vCloud Networking and Security Edge instance. This is used to create a load balancing rule that leverages the vCloud Networking and Security Edge DNAT capabilities.

7. Confirm that the virtual server was created as intended.



At this point, connecting from outside the organization to http://192.168.0.125 results in the vCloud Networking and Security Edge Gateway balancing in round-robin the two Web servers with IP addresses http://10.179.3.11 and http://10.179.3.11 respectively.

This load balancing configuration was done at the vCloud Networking and Security Edge Gateway level, backing the organization network. This configuration is not possible on the vCloud Networking and Security Edge device backing a vApp network.

4.2 Static Routing

Deployment Models: private, public.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

4.2.1 Background

The static routing functionality of vCloud Networking and Security Edge (Edge) that is exposed through vCloud Director can be used to route packets and establish communication between vApp networks that would not normally be able to communicate. The two vApp networks can be connected to the same the virtual datacenter network or to separate organization virtual datacenter networks. While establishing an inter-organization IPsec VPN tunnel between the QE and Engineering organization would provide connectivity, using the static routing feature limits communication to specific vApp networks defined in the routing statements. This example covers specifically static routing and not NAT using Edge firewall rules to provide security and limit access to the destination IP address.

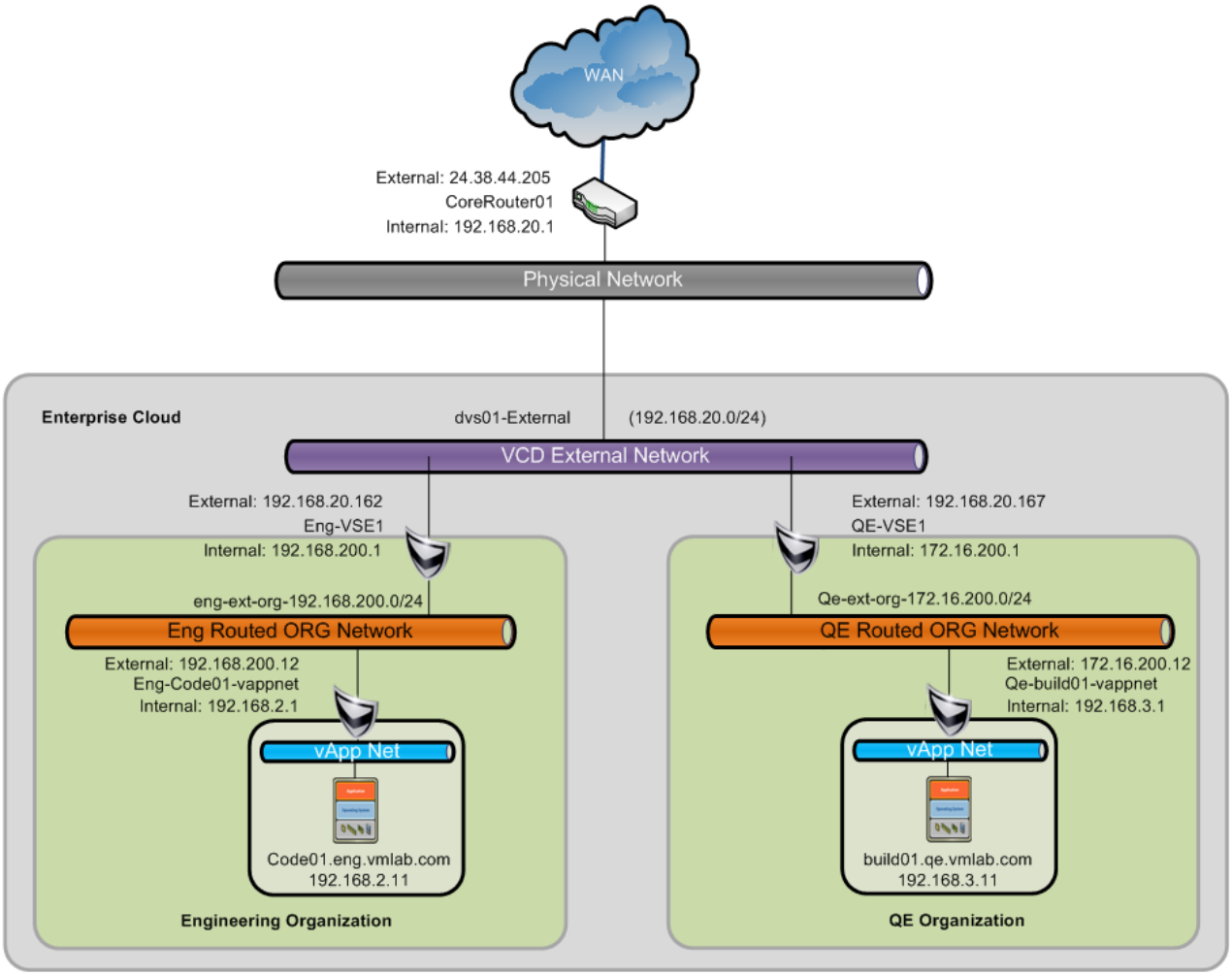
4.2.2 Use Case

In this static routing example, the build01.qe.vmlab.com virtual machine deployed in the QE organization must access a specific code repository server code01.eng.vmlab.com that exists in a separate engineering organization for periodic downloads of the updated software builds for testing. This testing takes place on virtual machines inside the QE organization and the communication remains inside the internal physical network.

4.2.3 Example

The following figure shows an example of static routing.

Figure 8. Routing Example Logical Architecture



Notes

- The implementation steps assume that both the Engineering and QE organizations have been provisioned with a single external routed organization virtual datacenter network in each organization. The creation of these two organization networks creates two vCloud Networking and Security Edge (Edge) devices in the System-virtual datacenter.
- Disable the firewall service in all four Edge devices below to allow for testing of communication between vApp networks. Normally you would Enable the firewalls and use the Default Deny policy coupled with Allow policies for the specific traffic patterns that are required.

Table 10. Network Device Information

Device	Location	IP Address	Notes
Corerouter01	Physical network perimeter	192.168.20.1 24.38.44.205	Cisco ASR No static routes defined Gateway for physical network to Internet
QE-vse1	VCD system virtual datacenter	Internal: 172.16.200.1 External: 192.168.20.167	vCloud Networking and Security Edge 5.1 QE external organization network
Eng-vse1	VCD system virtual datacenter	Internal: 192.168.200.1 External: 192.168.20.162	vCloud Networking and Security Edge 5.1 Engineering External organization network
QE-build01-vappnet	VCD system virtual datacenter	Internal: 192.168.3.1 External: 172.16.200.12	vCloud Networking and Security Edge 5.1 qe-build01 routed vApp network
Eng-code01-vappnet	VCD system virtual datacenter	Internal: 192.168.2.1 External: 192.168.200.12	vCloud Networking and Security Edge 5.1 eng-code01 routed vApp network
Build01.qe.vmlab.com	QE organization	172.16.200.11	Ubuntu 11.10

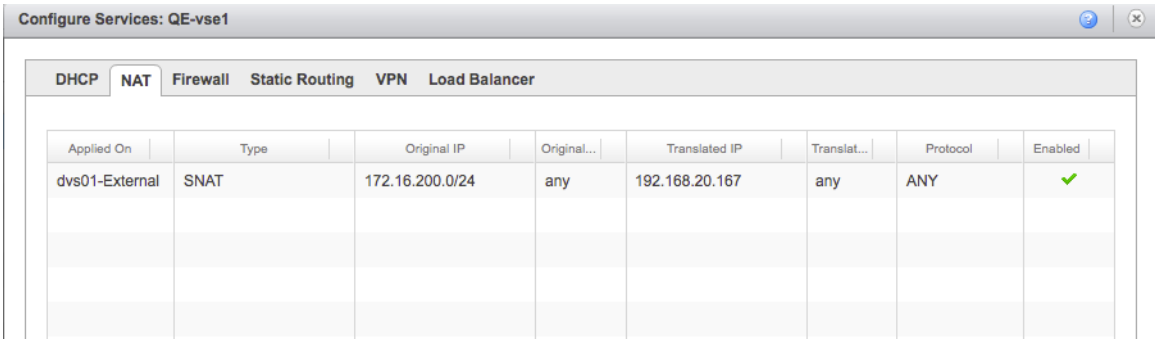
Code01.eng.vmlab.com	Engineering organization	192.168.200.11	Ubuntu 11.10

4.2.3.1. Organization Virtual Datacenter Network Configurations

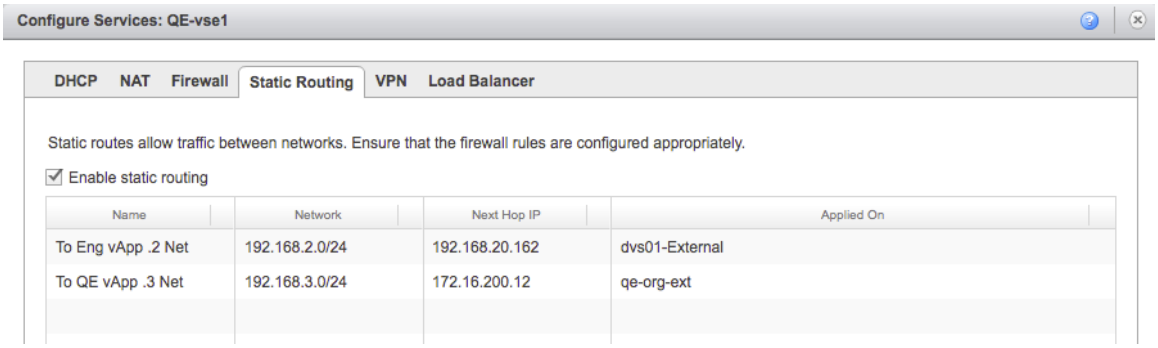
The configuration of static routes is performed on the **Gateway Services** tab of both the QE and Engineering organization virtual datacenter networks. A SNAT rule is created if it doesn't already exist on both the QE and Engineering organization virtual datacenter networks. Two static routes are created on both organization virtual datacenter networks. These two routes correspond to the destination or external vApp network and the source or internal vApp network.

To configure static routes

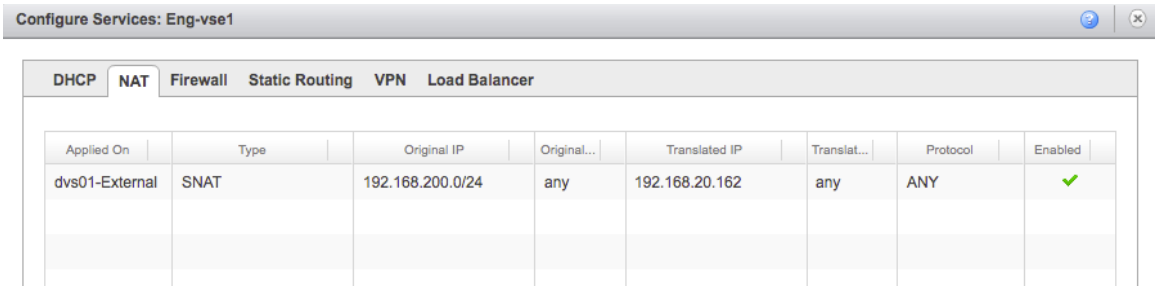
QE Organization Virtual Datacenter Network Gateway Services – NAT



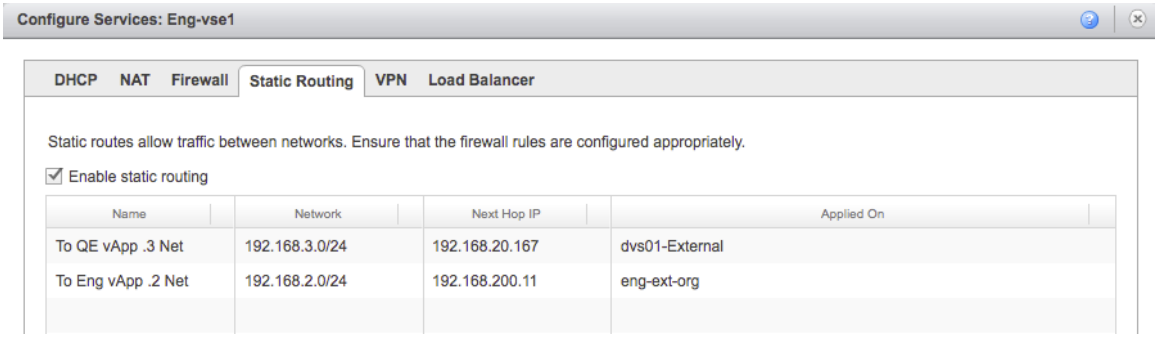
QE Organization Virtual Datacenter Network Gateway Services – Static Routing



Engineering Organization Virtual Datacenter Network Gateway Services – NAT



Engineering Organization Virtual Datacenter Network Gateway Services – Static Routing

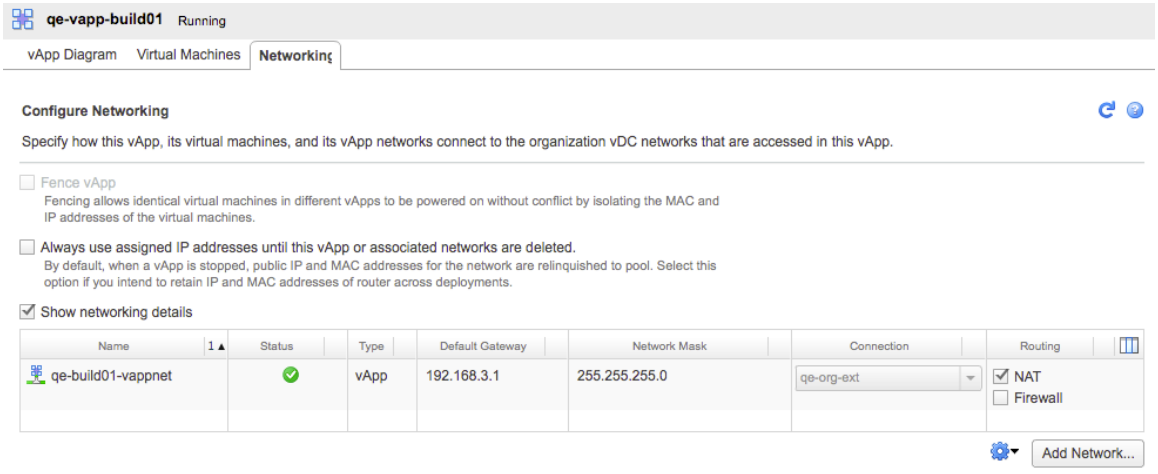


4.2.3.2. vApp Network Configurations

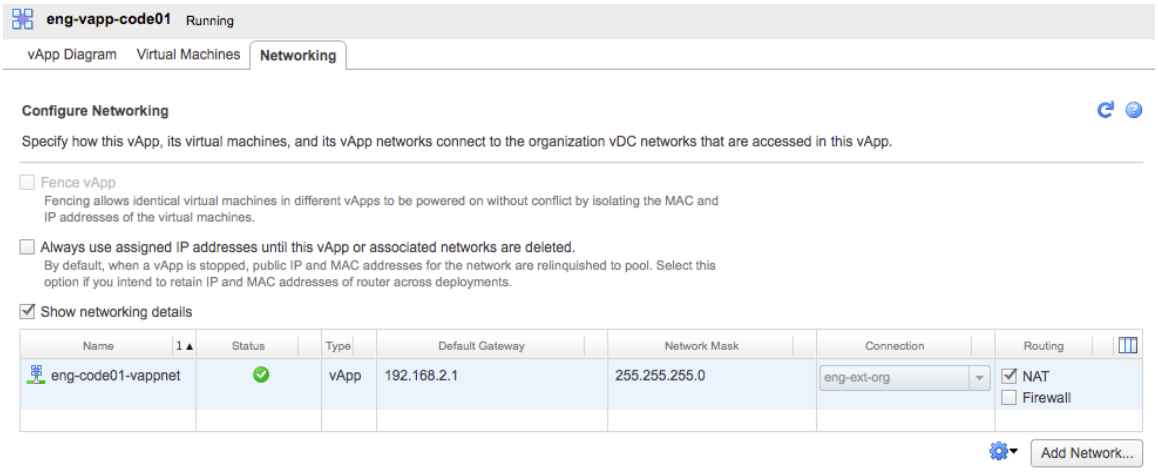
For both of the vApps, create a routed vApp network connected to the parent external organization network and then disable the firewall for testing. vCloud Director deploys a vCloud Networking and Security Edge device for these networks. There is no further configuration needed on these vApp networks, because the routing configuration is all performed on the external routed organization networks in the previous steps.

To create a routed vApp network

qe-vapp-build01 vApp Networking



eng-vapp-code01 vApp Networking



4.2.4 Design Implications

- Use static routing when you need a specific connection between vApp NAT networks.
- Configure a static route in vCloud Director when there is an existing IPsec VPN. This is handled automatically by the VPN as the target subnet is deemed “Interesting Traffic” and automatically sent over the VPN to the target network.

4.3 vCloud Networking and Security Edge Gateway Setup

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

4.3.1 Background

As of version 5.1, the CIS stack vCloud Networking and Security Edge (Edge) instance (or vCloud Networking and Security Edge Gateway) is an integral part of the virtual datacenter construct in vCloud Director. (In earlier versions, an Edge instance and its associated networks were treated as organization objects.)

4.3.2 Example

In this example a vCloud administrator provisions a vCloud Networking and Security Edge Gateway as part of an organization virtual datacenter provisioning process. This Edge instance is connected to an external network and an internal organization virtual datacenter network.

Later this example demonstrates how an organization administrator can deploy an additional internal organization virtual datacenter network and connect it to the same vCloud Networking and Security Edge Gateway.

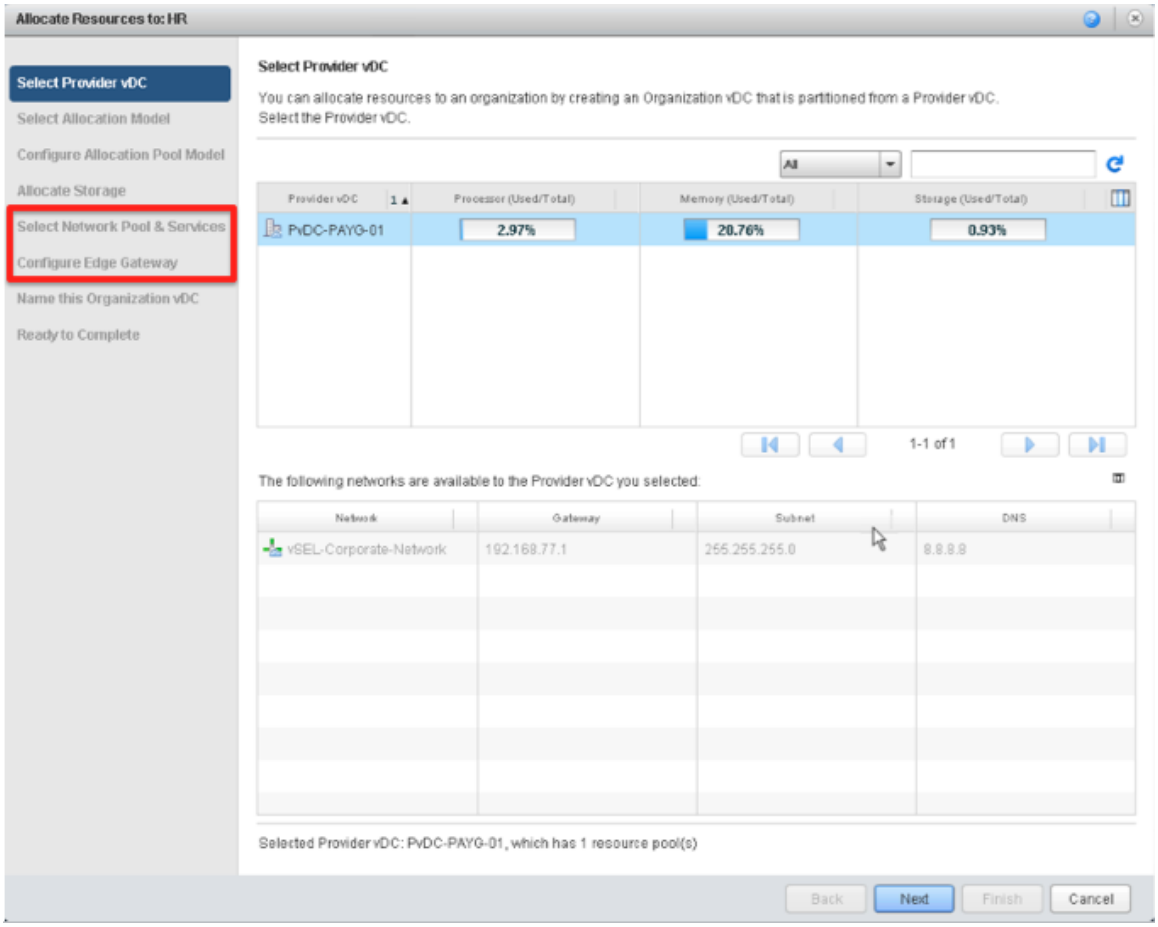
4.3.2.1. vCloud Networking and Security Edge Deployment and Configuration by the vCloud Administrator

The vCloud administrator can deploy a vCloud Networking and Security Edge Gateway at various stages. For example, a vCloud Networking and Security Edge Gateway can be provisioned while creating a new virtual datacenter (because the vCloud Networking and Security Edge object is now part of the virtual datacenter). A vCloud Networking and Security Edge Gateway can also be added later to an existing virtual datacenter but this is out of scope for this example.

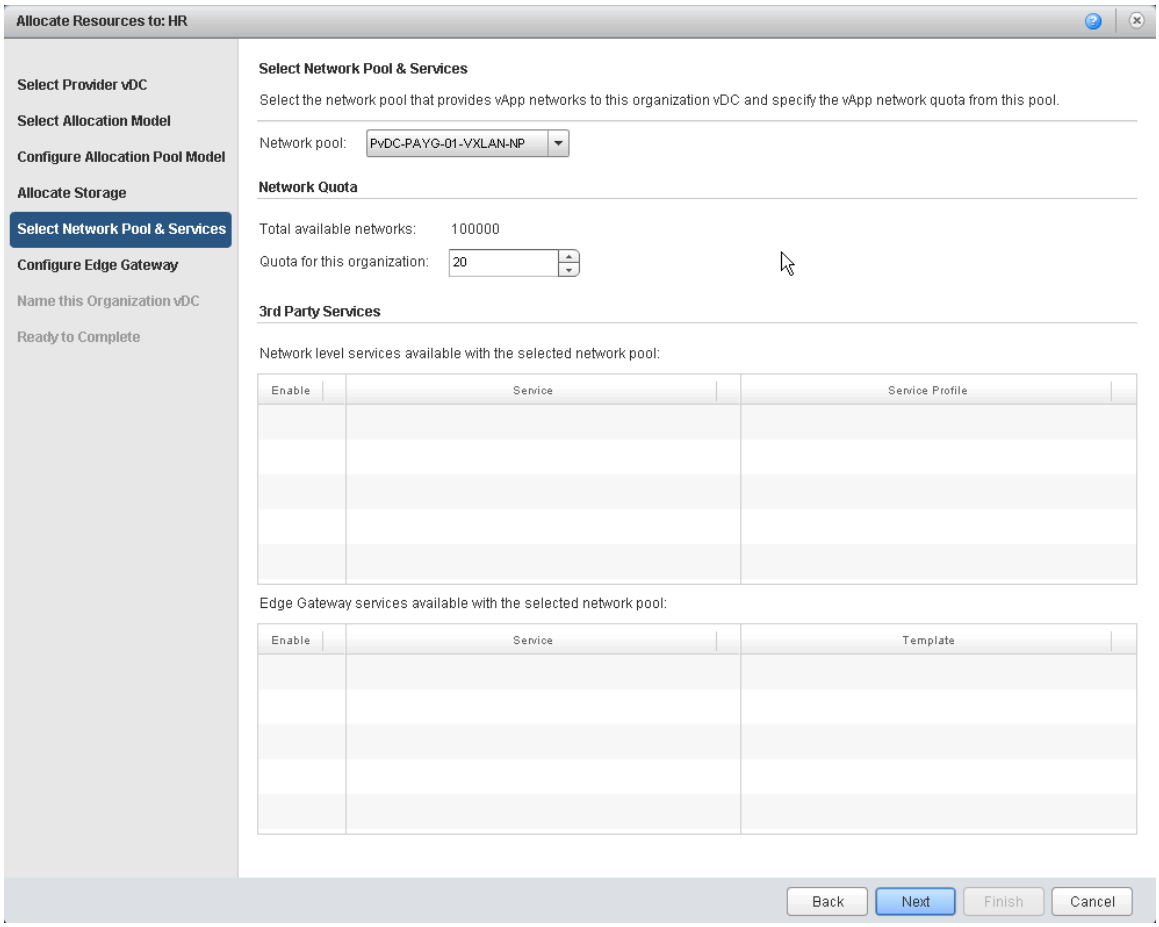
In the following procedure, the initial page of the wizard is used to create a virtual datacenter. The steps in the screenshots for the wizard are highlighted in red.

To use the wizard to create a virtual datacenter

1. On the **Allocate Resources** screen, click Select **Network Pool & Services**. Click **Next**.



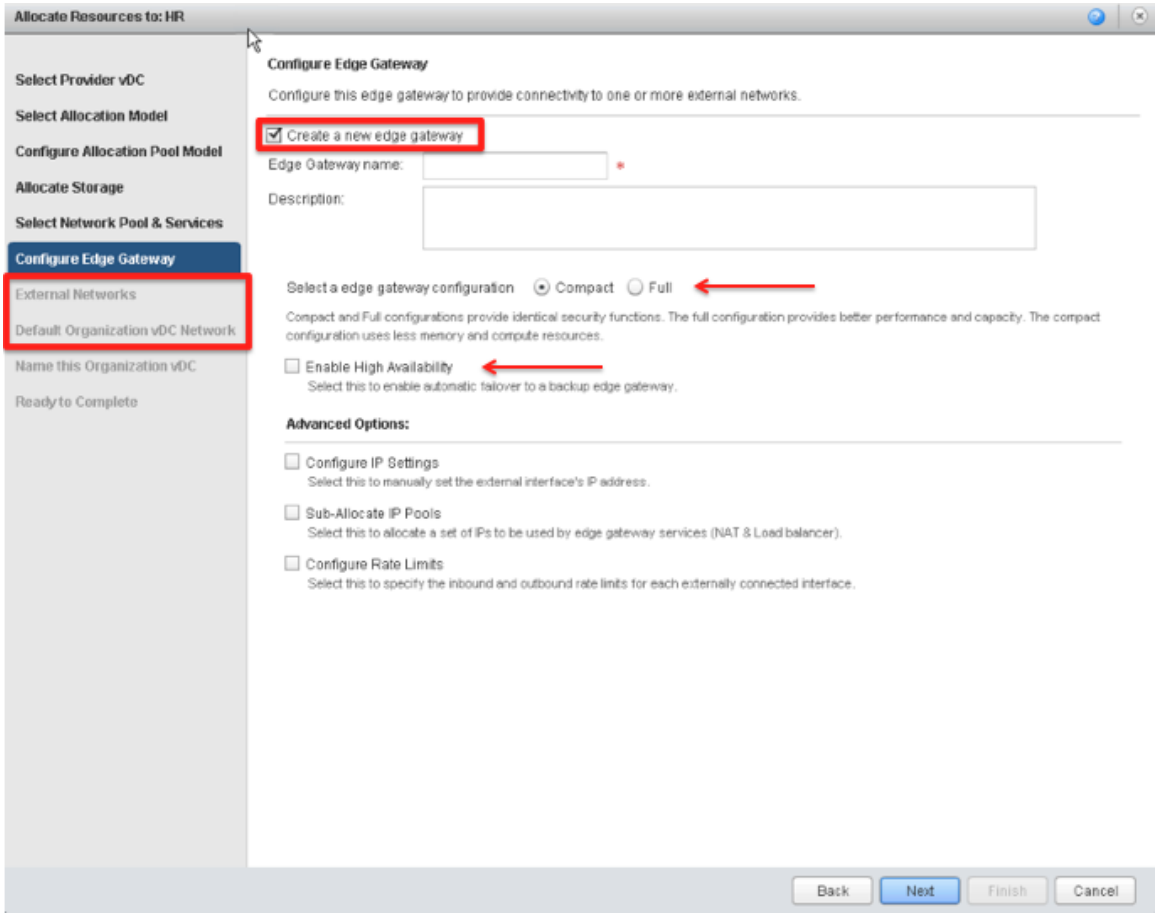
2. As part of this process the vCloud administrator entitles the organization for a certain number of networks. In the following example, the HR organization is entitled for 20 networks out of an existing VXLAN network pool.



Note: Before vCloud Director 1.5, this parameter was used only to limit the number of vApp networks an organization could create. As of vCloud Director 5.1 this number entitles and limits an organization for both vApp networks as well as organization virtual datacenter networks to be attached to the vCloud Networking and Security Edge Gateway being created. In fact, now an organization administrator can create organization virtual datacenter networks in self-service mode.

Click **Next** to proceed to the next screen.

3. The vCloud Networking and Security Edge Gateway wizard page is displayed and the vCloud administrator responds whether an Edge Gateway must be deployed. If **Create a new edge gateway** is selected, the **Configure Edge Gateway** screen is displayed.



Note that the number of provisioning steps in the left pane has increased. This is because the provisioning process must accommodate additional information associated to the Edge.

Make appropriate choices for **Select a edge gateway configuration** (**Compact** or **Full**) and **Enable High Availability** (selected or deselected).

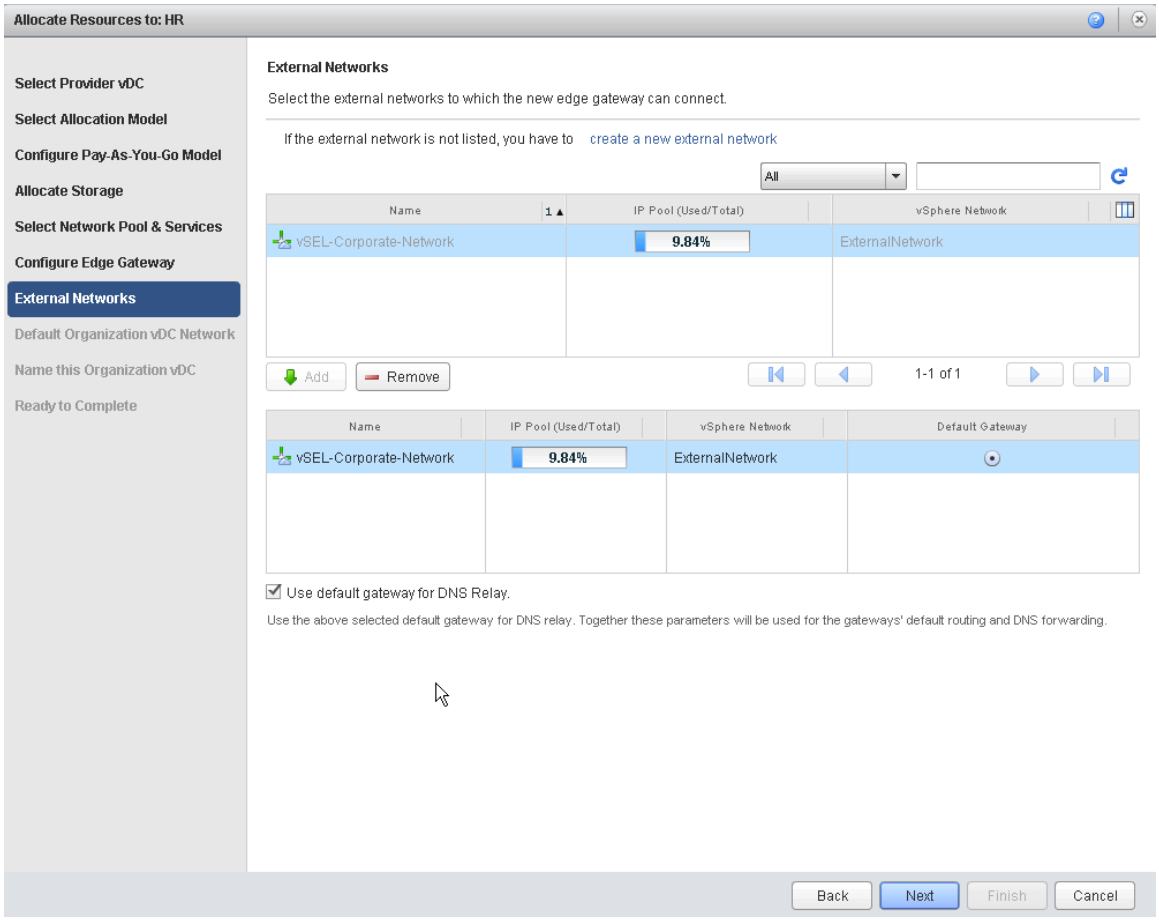
In vCloud Director 5.1, *Compact Edge* and *Full Edge* were introduced. These are two different vCloud Networking and Security Edge Gateway virtual machine configurations that provide different input/output throughput. These configurations are related to different virtual hardware configurations as well as different parameters inside the vCloud Networking and Security Edge (Edge) software stack.

Similarly, *Edge HA* is a VCD 5.1 resiliency feature. If HA is enabled, vCloud Director and vCloud Networking and Security Manager deploy two Edge devices in a clustered configuration. Edge previously leveraged the traditional vSphere HA technology to provide resiliency. If the physical server running the single Edge instance failed, vSphere HA would restart the Edge virtual machine on another server. This means that the associated VCD organization would not be able to communicate externally until the same Edge instance is restarted on a different physical server. With Edge HA introduced in vCloud Director 5.1, the two virtual machines work as a pair and can fail over immediately.

Additional advanced features can be selected in this page. If selected, an additional configuration page is added in the provisioning wizard.

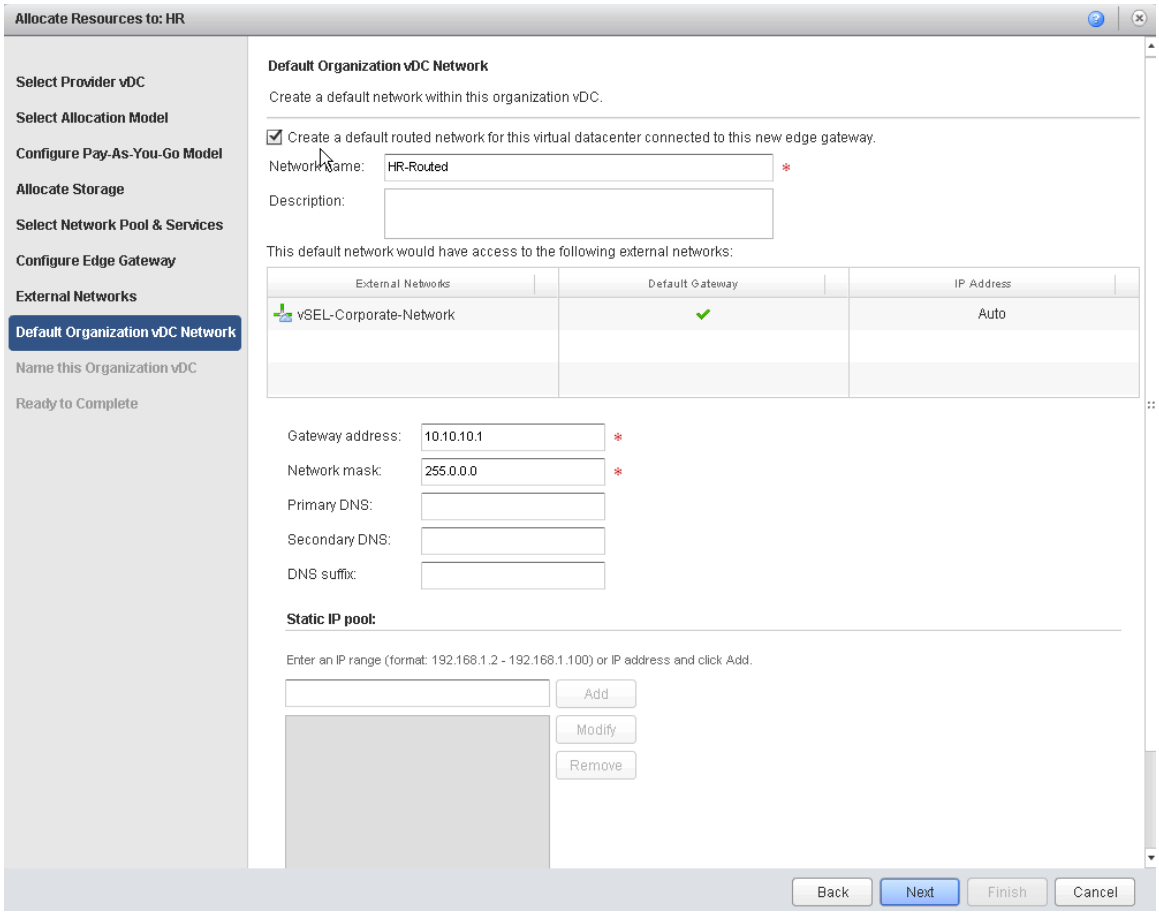
Click **Next** to proceed to the next screen.

4. Choose an external network and click **Next**.



Note: Beginning with vCloud Director 5.1, more than one external network can be selected. This is different from the earlier version where only one external network and one organization network could be selected. In this example there is only one external network, so only one can be selected. Now, the vCloud Networking and Security Edge Gateway can be set to act as a DNS relay.

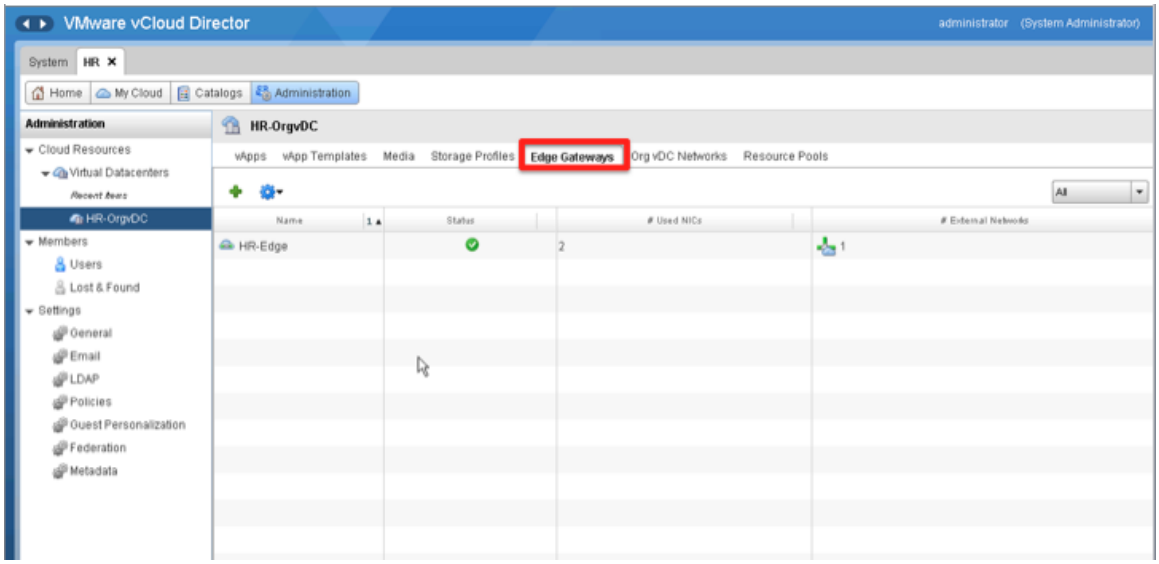
5. Create an organization virtual datacenter network. If the vCloud administrator directs the wizard to create a network, the following page is presented. After completing this screen, click **Next**.



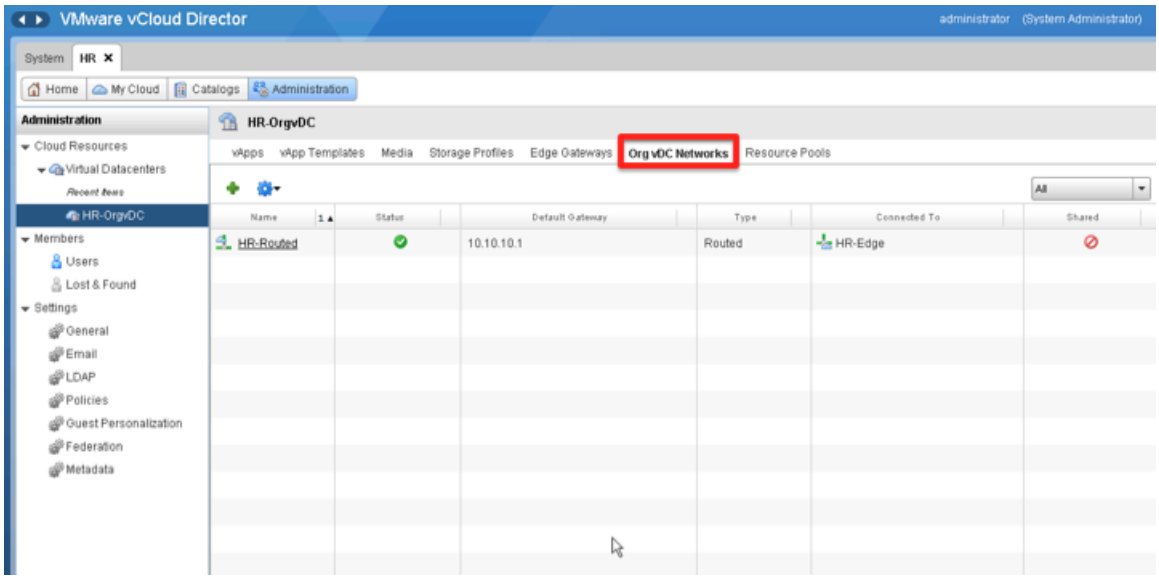
The network is named “HR-Routed.” It is the only network currently available in the organization.

Upon successful completion of the wizard, the resources are available to the organization. The vCloud Networking and Security Edge Gateway, along with the organization virtual datacenter networks, are all integral parts of the virtual datacenter.

The following screenshot shows the **Edge Gateways** tab.

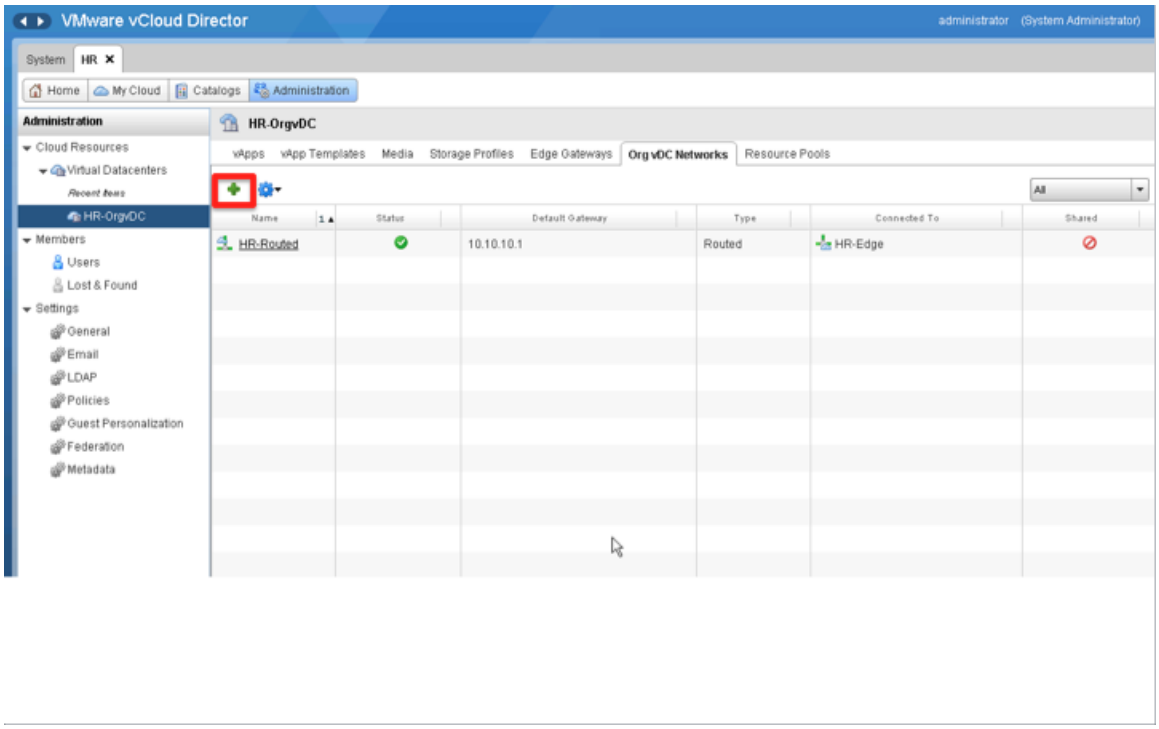


The following screenshot shows the **Org vDC Networks** tab.

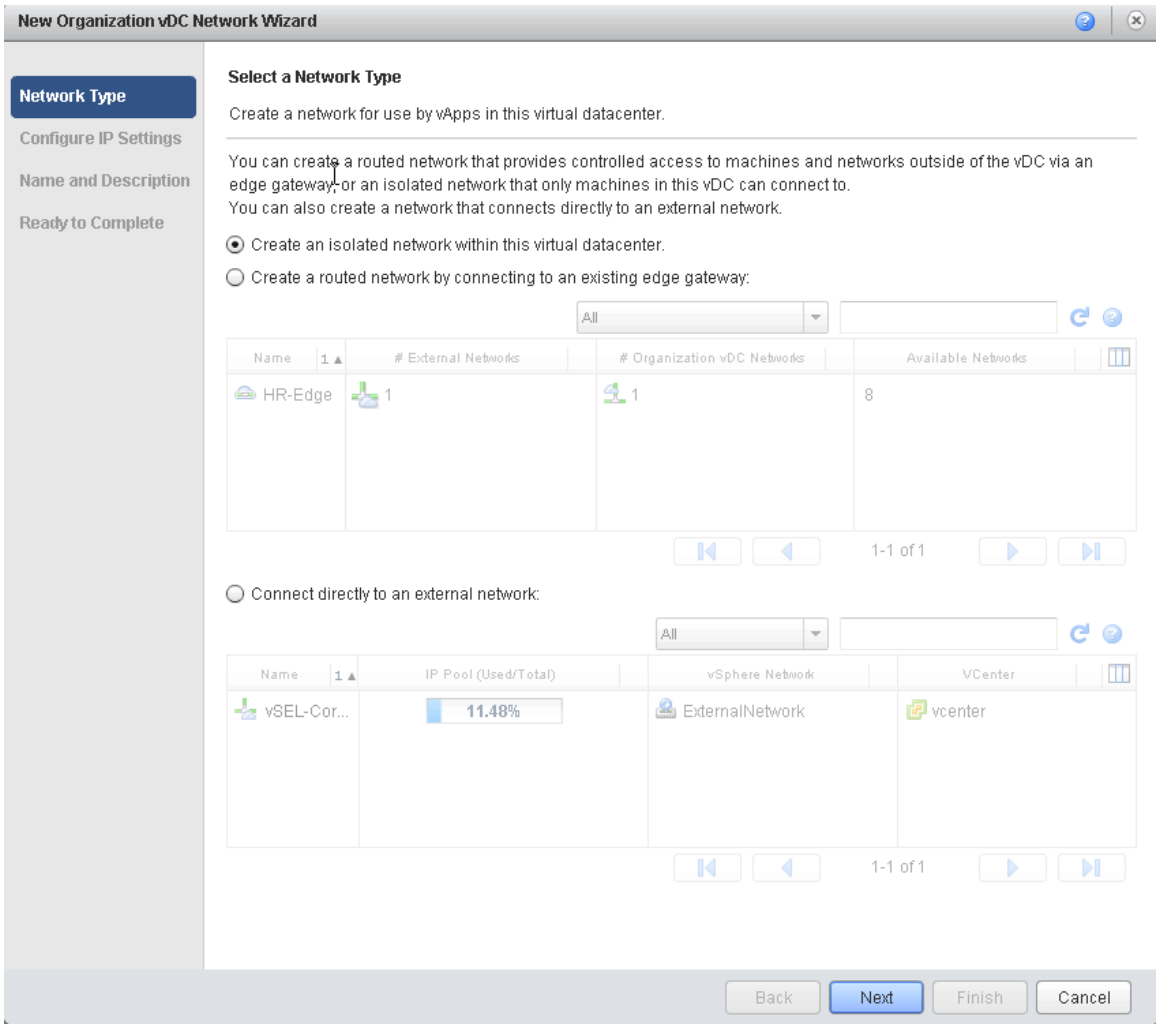


What has been shown so far is how to provision a vCloud Networking and Security Edge Gateway (Edge Gateway) as part of the virtual datacenter provisioning wizard. The vCloud administrator can also create the Edge Gateway (or add an additional Edge Gateway) by clicking **Add Gateway** from the **Edge Gateways** tab of the organization virtual datacenter consolidated view.

6. Similarly, a vCloud administrator can also add additional organization virtual datacenter networks in the organization virtual datacenter by clicking the green plus sign (Add Network) in the **Org vDC Networks** tab.



7. Click the link to open a wizard that allows the creation of a new network inside the virtual datacenter. The following screenshot shows what is displayed for a vCloud administrator when adding a network to the virtual datacenter.



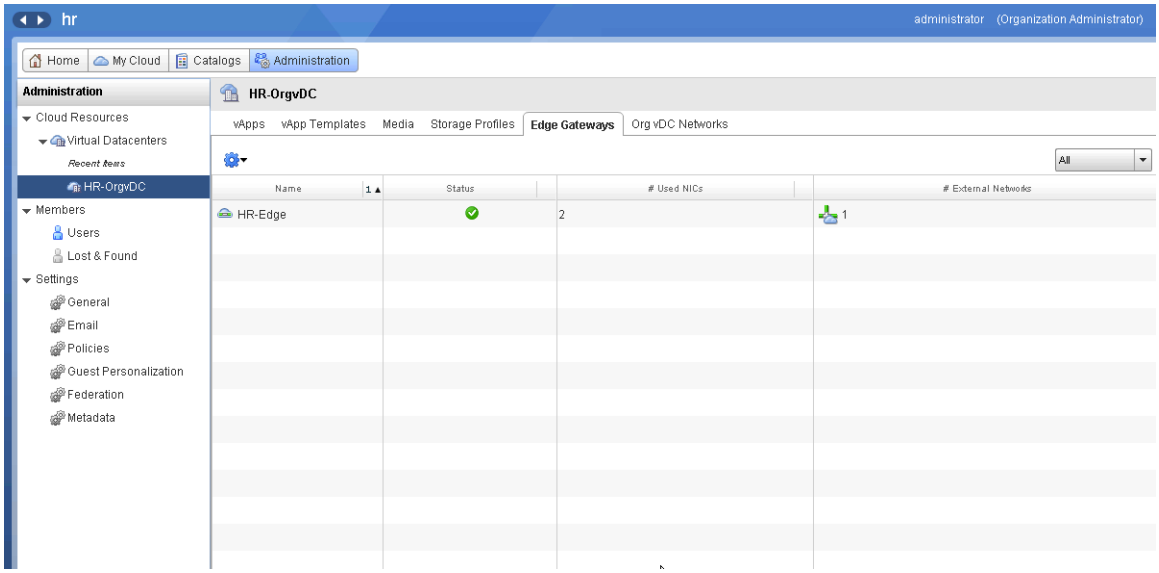
The vCloud administrator can create all three types of networks, including a direct connect to the external network (bypassing the Edge Gateway). This is not an option for the organization administrator. The organization administrator *cannot* deploy an additional vCloud Networking and Security Edge device from the **Edge Gateways** tab.

4.3.2.2. vCloud Networking and Security Edge Configuration and Deployment by the Organization Administrator

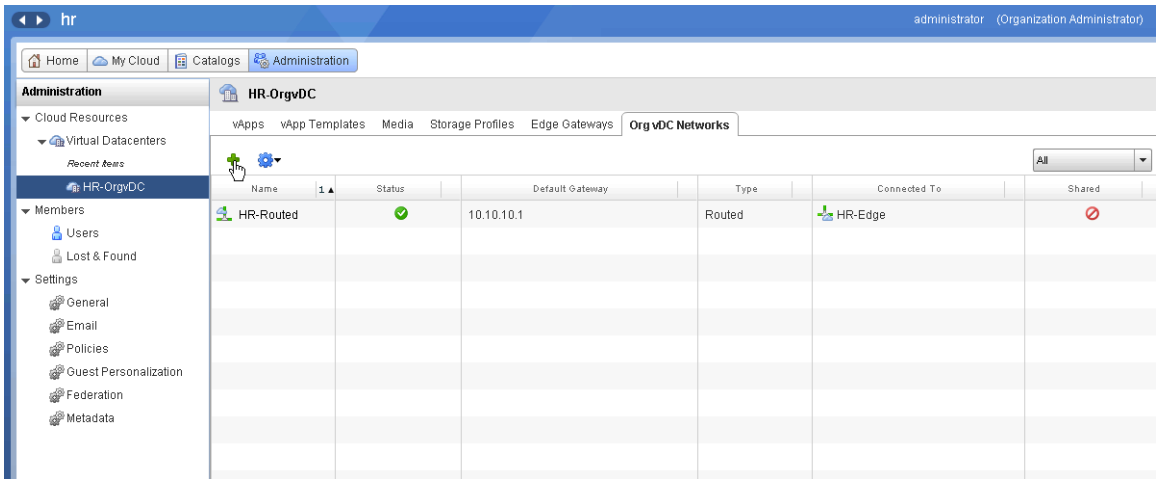
In this session, the HR organization administrator takes over from where the vCloud administrator left off. The organization administrator's view of the organization virtual datacenter is similar to that of the vCloud Administrator.

To configure and deploy the vCloud Networking and Security Edge instance

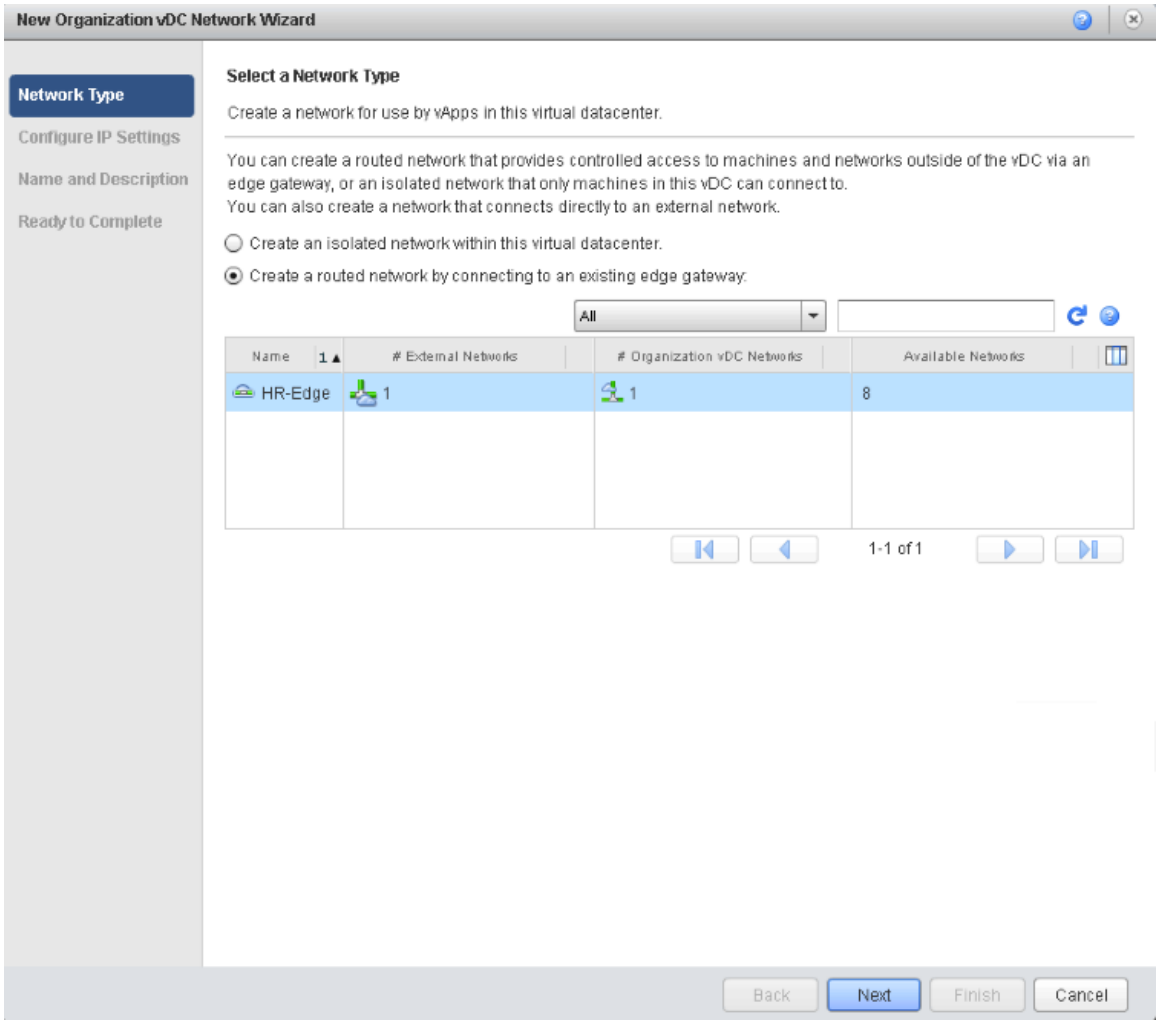
1. On the **hr** screen, select **HR OrgvDC** and click the **Edge Gateways** tab.



2. Under the **Org vDC Networks** tab, add a network by clicking the green plus sign (Add Network).



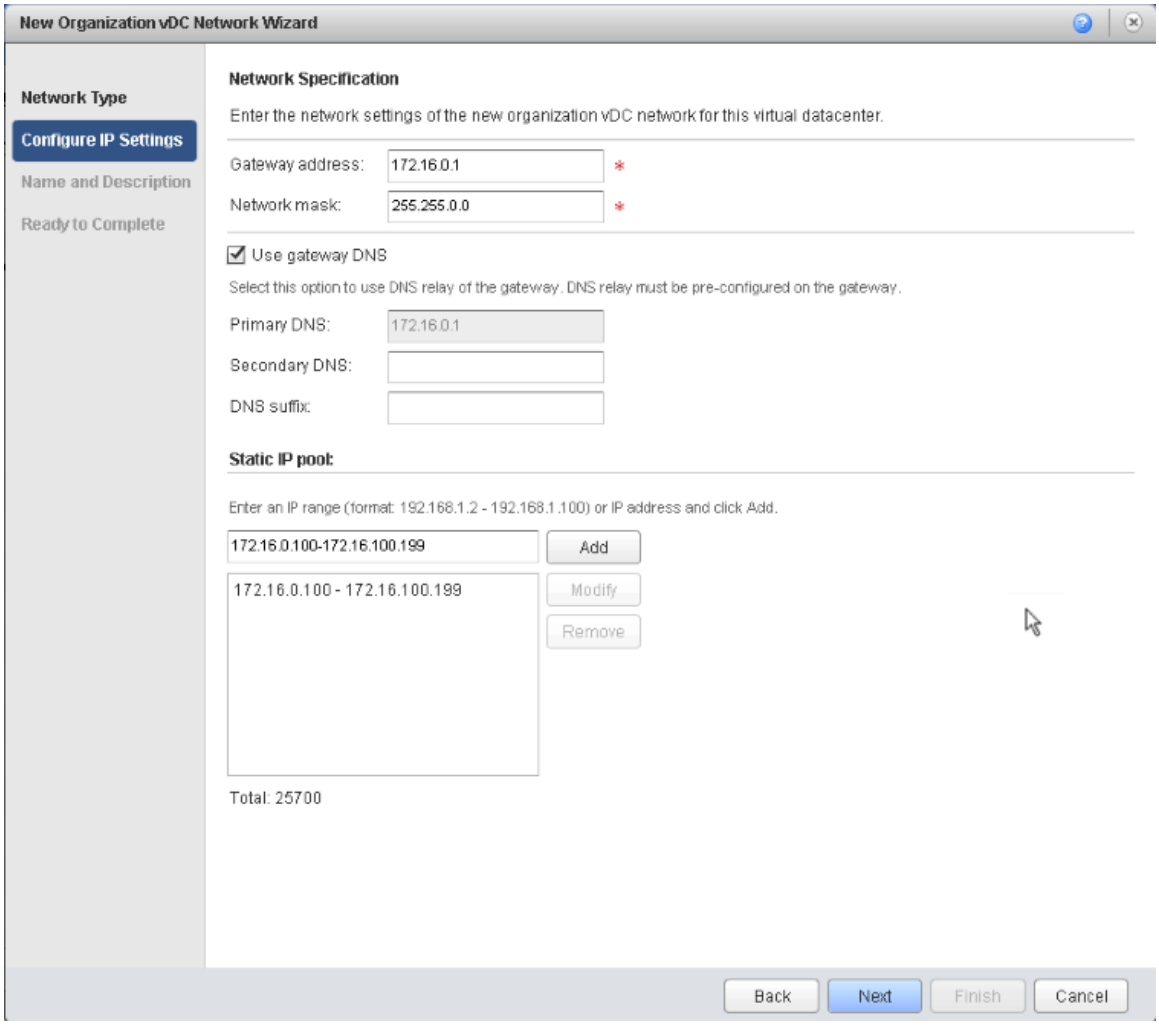
3. The **New Organization vDC Network Wizard** is displayed. After completing this screen, click **Next**.



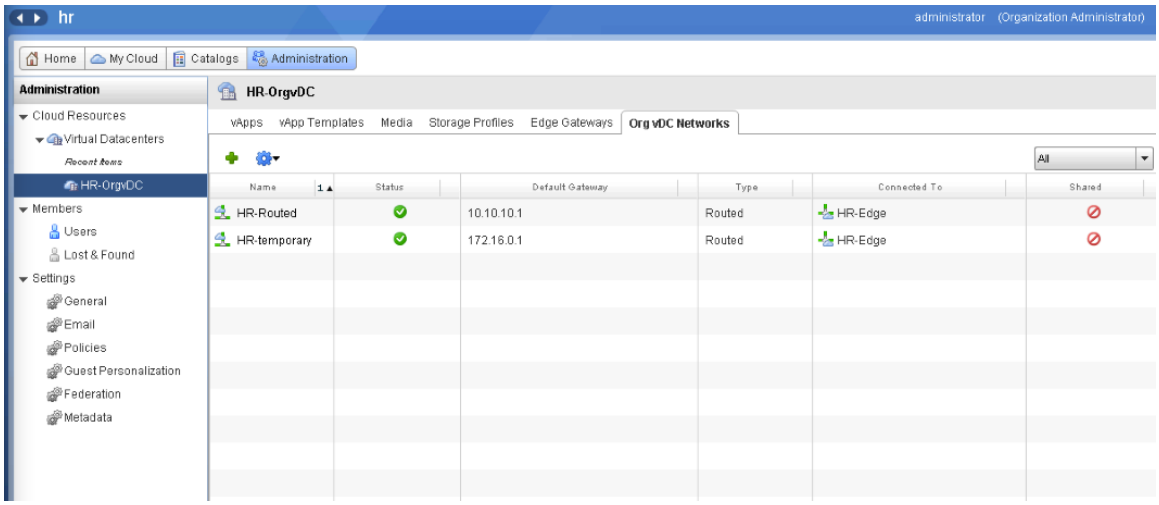
The organization administrator cannot create direct connections to external networks.

An organization administrator can create an isolated or routed network and connect it to an existing vCloud Networking and Security Edge Gateway. In this example we only have one vCloud Networking and Security Edge (Edge) instance, so we will create a “temporary” network and connect it to this existing Edge instance.

4. On the **Network Specification** screen, define and personalize the new network. Click **Next**.



5. Set the name for the new network. In this example it is named **HR-temporary**. The new network is displayed on the summary page of all the organization virtual datacenter networks.



Users in the HR organization can now attach virtual machines to both of these networks that are routed to the external network using the same Edge Gateway. The two organization virtual datacenter networks can also access each other using static routing configurations automatically defined on the single Edge Gateway.

There is no longer a need to create an Edge device for each routed network deployed. Also, beginning with vCloud Director 5.1, the organization administrator can create, in self-service mode, organization virtual datacenter networks.

Additionally, the organization administrator can configure all of the possible Edge Gateway services such as DHCP, NAT, firewall, static routing, VPN, and load balancing.

4.3.3 Design Implications

A vCloud Networking and Security Edge Gateway (either compact or full) can support a maximum of total 10 networks in any combination of external networks and organization virtual datacenter networks.

4.4 Public vCloud External Network

Deployment Models: public.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

4.4.1 Use Case

An external network is a method of providing communication to resources outside of a vCloud. In vCloud Director the external network is a logical representation that maps 1:1 to an existing vSphere port group on a standard or distributed virtual switch.

4.4.2 Example

In this external network example, the service provider uses existing network automation software to dynamically provision the vSphere and corresponding vCloud Director networks. The service provider uses an automation platform to dynamically automate the following tasks during customer onboarding:

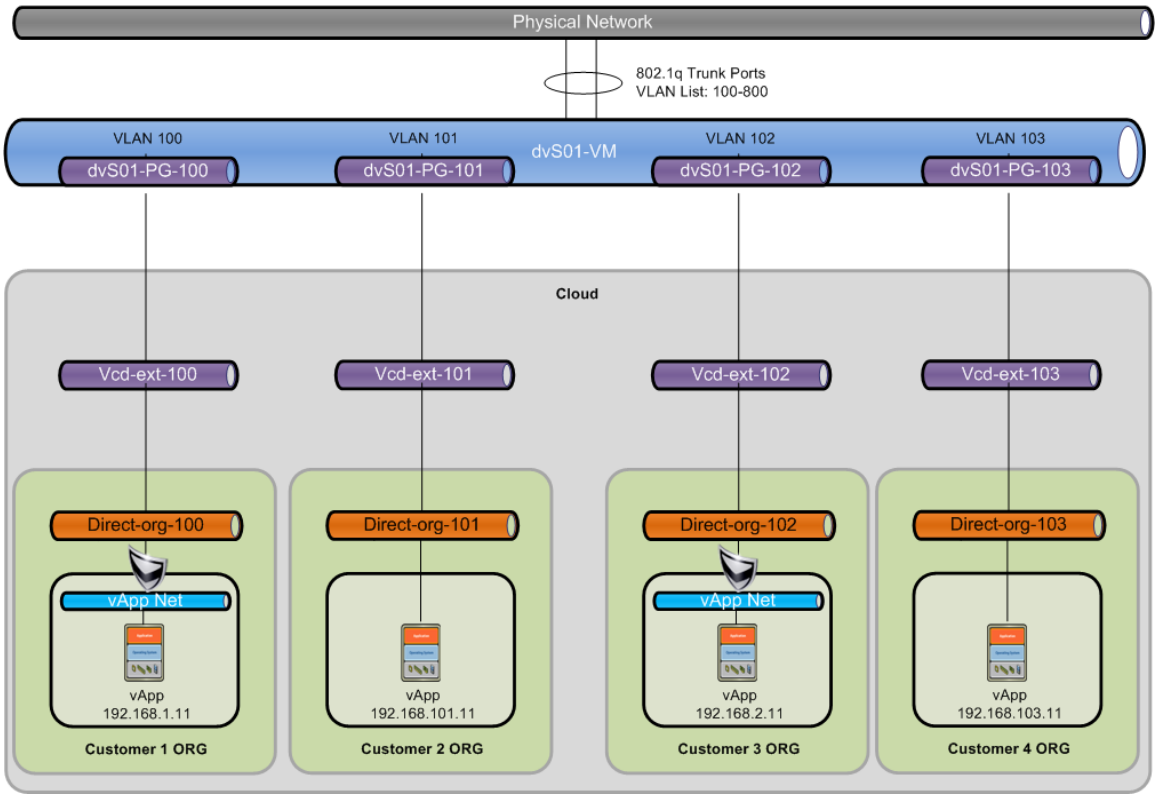
- Provision a vSphere port group on the dvSwitch for each customer.
- Assign an appropriate VLAN to this port group for each customer.

  Note: The external network can be differentiated through use of a separated physical network or VLAN. If using VLANs, only a single VLAN can be used on the port group.
- Provision a dedicated vCloud Director external network and map it to the port group created in step 1 for each customer.
- Create a direct connect external organization network for each customer.

Figure 9 shows a VLAN configuration that uses 802.1q VLAN trunk ports on the physical switches to the ESXi dvSwitch uplinks. This enables the physical switching infrastructure to allow all the VLANs configured in the infrastructure to communicate to the ESXi hosts while still keeping the VLANs separated and in separate broadcast domains. The dvSwitch delivers the appropriate Ethernet frames to the appropriate port group based on a match of the VLAN tag on the frame and the VLAN associated with the port group. The dvSwitch port groups remove the VLAN tag from the Ethernet frame and deliver it to the appropriate virtual machine. This architecture is commonly referred to as *Virtual Switch Tagging* (VST).

In the figure, four organizations are shown, two of which have vApps direct connected to the parent organization network, and two of which have a vApp network connected to the parent organization network.

Figure 9. Service Provider External Network Example



The vSphere configuration to support this architecture requires separate dvSwitch port groups for each customer and a VLAN provisioned for each. Figure 10 and Figure 11 show four customers configured in this environment.

Figure 10. vSphere Port Group Configuration

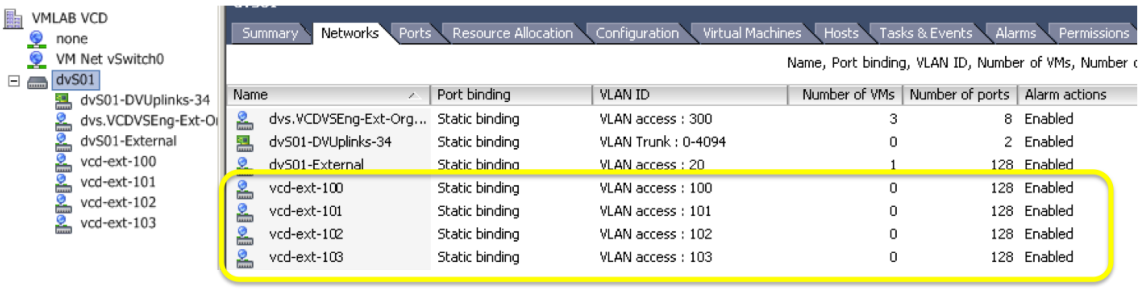


Figure 11. vCloud External Networks

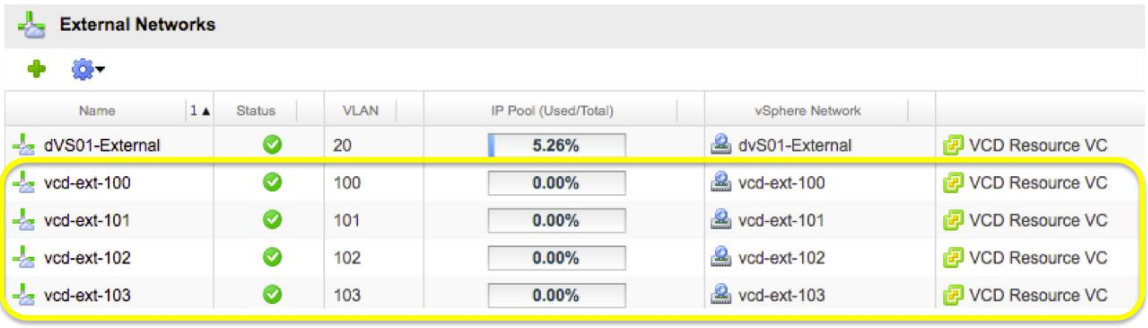


Figure 12 and Figure 13 show the network specification for one of the customer vCloud external networks (vcd-ext-101). A network specification represents a subnet and its associated configuration for the external network.

Figure 12. vcd-ext-101 External Network Configuration

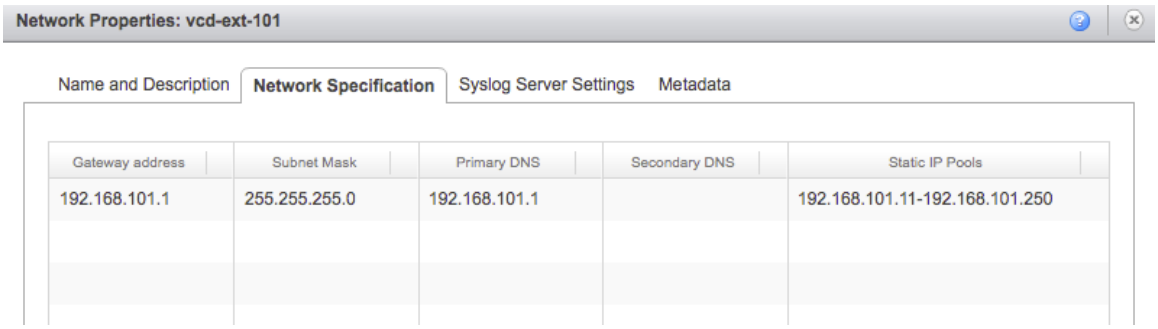
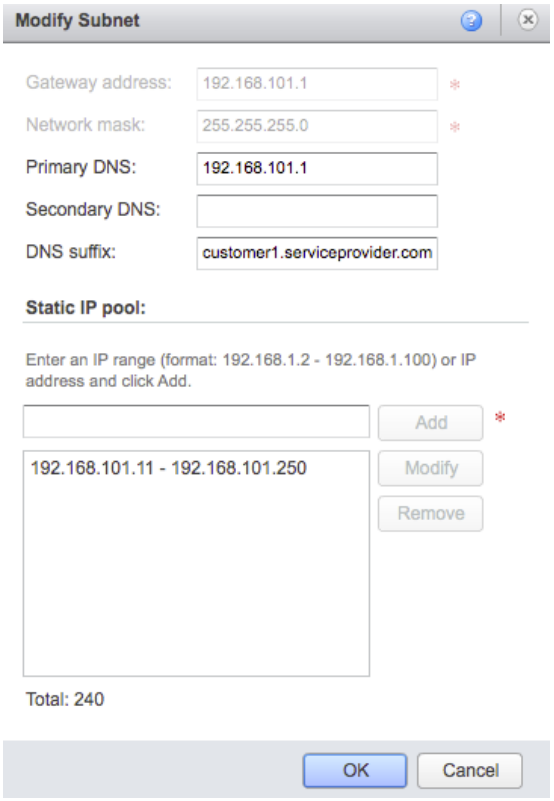


Figure 13. Network Specification Properties



In this example, a static IP pool was configured providing a total of 240 IP addresses. vCloud Director allows multiple static IP pools for each external network. These addresses can be used for assignment by vCloud Director to virtual machines or external interfaces of the vCloud Networking and Security Edge devices. The gateway address in this configuration is 192.168.101.1, which is a logical interface on the Cisco Layer 3 switching infrastructure.

4.4.3 Management

As the vCloud Director environment grows, the service provider can modify the external network settings.

To modify external network settings

1. Add a new network specification (subnet) to an existing external network.
2. Modify an existing network specification (DNS servers or suffix) for the external network. You cannot modify the network subnet mask or default gateway.
3. Add or remove IP addresses in the static IP pool for an external network.

Note: After you have created a network specification (subnet) for an external network, you cannot delete it.

4.4.4 Design Implications

- This service provider has been providing managed compute and networking services for years developing an established process for datacenter automation. The advantage for the service provider is that the established process and automation technology can be used as a foundation.
- Although external networks can be shared between multiple organizations and organization virtual datacenter networks, service providers will be more likely to dedicate an external network for each customer.

By leveraging the existing investments and established process around datacenter automation, the service provider is automating what vCloud Director would typically handle at the networking layer. vCloud Director uses network pools to provide Layer 2 isolation of a tenant's virtual machines from the vApp through the physical network using VLANs, VCDNI, or VXLAN.

4.5 VXLAN ORG Network for Disaster Recovery

Deployment Models: private, hybrid.

Software Components: vCloud Director 5.1, vCloud Networking and Security Manager 5.1, vSphere 5.1.

Hardware Components: NETGEAR GS724T Switch, Sophos UTM Router.

4.5.1 Background

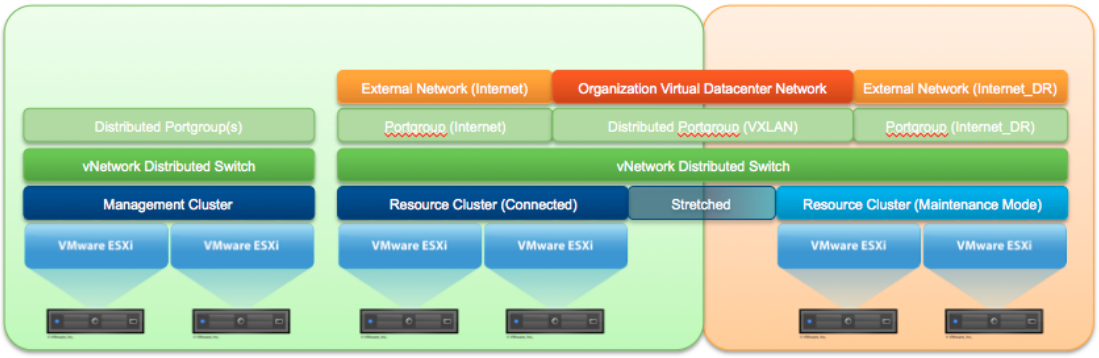
The process of failing over the management components or vApps from a primary vCloud to a disaster recovery site is documented in the *VMware vCloud Director Infrastructure Resiliency Case Study* (http://www.vmware.com/files/pdf/techpaper/vcloud-director-infrastructure-resiliency.pdf) In cases where stretched Layer 2 networks are present, the recovery of vApps is greatly simplified because vApps can remain connected to the same logically defined virtual networks regardless of the physical location in which the vApps are running. In cases where there is not a stretched Layer 2 network, VXLAN enables simplified DR and implementations of vCloud Director that span multiple locations. This is achieved by creating a Layer 2 overlay network without changing the Layer 3 interconnects already in place. This example illustrates failover of an SRM-based vCloud Director implementation without the need to reassign IP addresses to the virtual machines, as well as the scripted changes that would need to be done to simplify the process.

4.5.2 Example

In keeping with the reference infrastructure and methodology defined in the *VMware vCloud Director Infrastructure Resiliency Case Study*, this example uses a cluster that has ESXi members in both the primary and the recovery site. The premise is that workloads run in the primary site where the ESXi hosts are Connected. At the recovery site, the ESXi servers are in maintenance mode, but configured in the same cluster and attached to all the same vSphere Distributed Switches (VDS). The solution approach considered within the following sections is developed on the basis of the *VMware vCloud Director Infrastructure Resiliency Case Study,* and the prerequisites it defines are applicable to this solution. The failover of a management cluster for a vCloud infrastructure, in the absence of stretched Layer 2 networking, is also contained within this document.

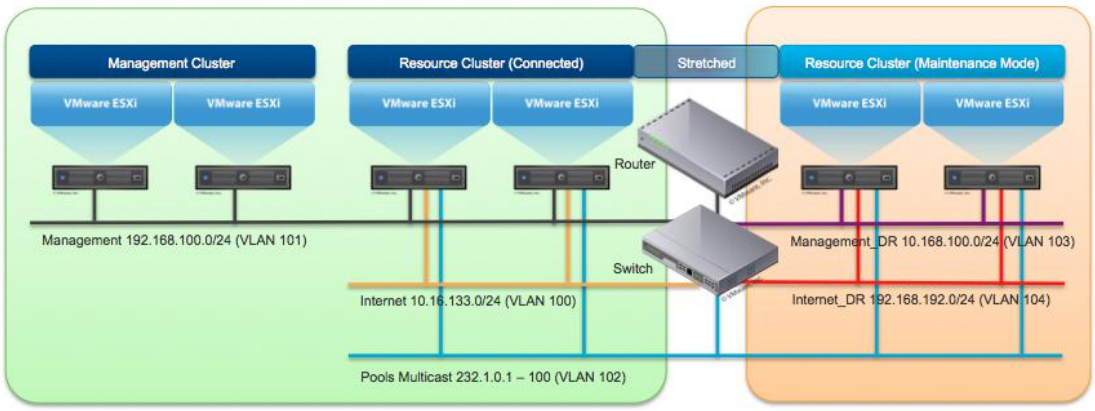
The following figure shows the logical architecture for this example.

Figure 14. Example Logical Architecture



All ESXi hosts in the resource cluster are connected to a common VDS with defined site-specific port groups for external networks, Internet and Internet_DR. In conjunction with this, an organization virtual datacenter network is defined and results in a port group from the VXLAN-backed network pool being deployed. The following figure shows the physical architecture for this example.

Figure 15. Example Physical Architecture

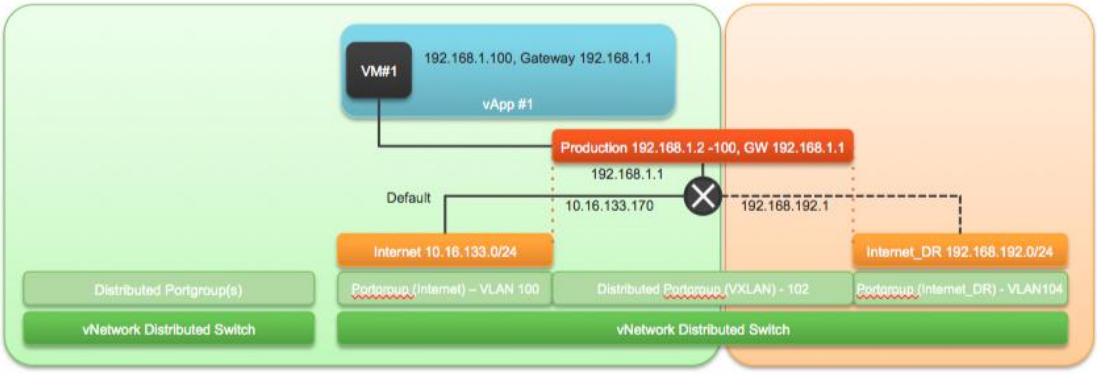


Note: For testing, a single switch and router/firewall were deployed to simulate the separate networks for the primary and recovery sites. Although this is not entirely consistent with a real world deployment, this configuration is representative for lab testing. The router shown in Figure 15 provides routing capability among all networks, with the exception of the pools network.

The ESXi hosts deployed in the production site are connected to a common Layer 3 management network. Similarly, the ESXi hosts deployed in the recovery site are connected to a common Layer 3 management network, but in a different Layer 3 than that of the network for the production site. In addition, the Internet external networks are the primary networks that will be used for vApp connectivity and are also in a different Layer 3 than the Internet network available at the recovery site. These are attached to vCloud Director as two distinct external networks.

vCloud Networking and Security Edge firewall rules, NAT translations, load balancer configurations, and VPN configurations must be reproduced on the DR side to maintain consistent configurations and make sure that everything will work after recovery. As shown in Figure 16, the example uses the vCloud API upon failover to duplicate the primary site configuration to the failover site. This eliminates much of the manual reconfiguration on the recovery side that would otherwise be required.

Figure 16. vCloud Director Network Configuration



The two Internet networks (Internet and Internet_DR) have been defined as external networks, with their respective IP configurations. In conjunction with this, a new organization virtual datacenter network (VXLAN-backed) called "Production" is defined. Finally, an Edge Gateway device is deployed (note the appliance is deployed in the Production site) with connectivity between the organization network and the two external networks. To facilitate virtual machine connectivity between the Production organization virtual datacenter network and the external network a number of destination NAT (DNAT) and source NAT (SNAT) rules are required. An example of these rules is shown in the following table.

Note: Although there is no technical reason for the Internt_DR DNAT rule to be disabled, the SNAT rule must be disabled so that network traffic is not inadvertently passed over the wrong interface to the Internet_DR network because it is not available in the production site.

Table 11. Sample NAT Rules

Applied On	Type	Original IP Address	Original Port	Translated IP Address	Translated Port	Protocol	Enabled
Internet	SNAT	192.168.1.0/24	*	10.16.133.171	*	TCP/UDP	Yes
Internet_DR	SNAT	192.168.1.0/24	*	192.168.192.2	*	TCP/UDP	No
Internet	DNAT	10.16.133.171	*	192.168.1.100	*	TCP/UDP	Yes
Internet_DR	DNAT	192.168.192.2	*	192.168.1.100	*	TCP/UDP	No

Note: An alternative to the chosen configuration is to implement a solution where the vCloud Networking and Security Edge Gateway is connected only to the active external network. It was decided to predefine the connections since this would present options for preconfiguring rules for the recovery site and thereby reduce reconfiguration steps during a recovery process.

During a vCloud DR process the expectation is that there is a requirement for the external IP addresses used to access the workloads to change to those used in the recovery site.

4.5.3 VXLAN Example Testing Summary

The following sections provide an overview of some of the testing conducted to validate the concept of using VXLAN to simplify vCloud DR recovery.

4.5.3.1. Test 1 – Prove Connectivity and Verify NAT Configuration

The purpose of this test is to verify that predefining connections to both the production and recovery sites is viable, and that following a disaster recovery scenario the required connectivity could be established to a recovered vApp. The following is the high-level test procedure.

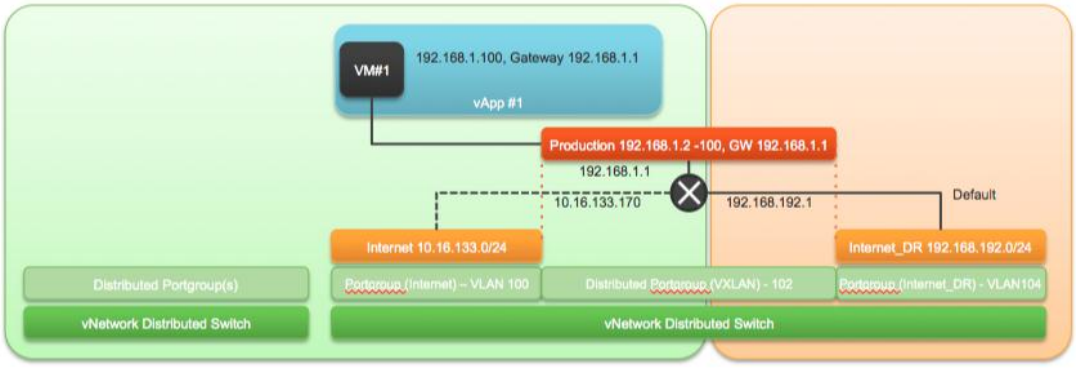
To prove connectivity and verify the NAT configuration

1. Validate connection to the vApp from a client device on the production external network 10.16.133.0/24 (SSH was used).
2. Change the default route defined on the vCloud Networking and Security Edge Gateway from the Internet network to the Internet_DR network and validate connectivity to the vApp. This uses the directly attached network so connectivity is maintained.
3. Fail over the vApp to the recovery site.
4. Validate connection to the vApp from a client device on the production external network 10.16.133.0/24. It fails, because the NAT addressing of the vApp is no longer connected to the directly attached vCloud Networking and Security Edge interface on the 10.16.133.0/24 network.
5. Enable the previously disabled SNAT/DNAT rules and validated connectivity from a client device on the failover external network 192.168.192.0/24 (Internet_DR) to the new address translated with NAT.
6. Remove the original Internet external network so that all connectivity is forced through the desired Internet_DR external interface on the vCloud Networking and Security Edge Gateway.
7. Validate connectivity from the vApp to the original client device on the 10.16.133.0/24 network (Internet) to the original client device (global routing between Internet and Internet_DR network should permit this to take place). This works because the Sophos UTM performs a NAT translation of the 192.168.192.0/24 to 10.16.1333.0/24 network on behalf of the vCloud Networking and Security Edge device.

During testing, the first three steps behave entirely as expected. Network traffic from the Internet network can successfully pass to the virtual machine defined in the vApp. Similarly, if the appropriate DNAT rule is enabled, network traffic can pass from the Internet_DR network.

When attempting to validate step 4, some interesting observations can be made. Despite reconfiguring the connection to the Internet_DR network as the default route, connections are still attempting to leave the vCloud Networking and Security Edge Gateway device over the original locally attached Internet interface, despite the default route being updated to go out the Internet_DR interface. If the client device is then connected to the Internet_DR network with an appropriate IP address, then network connections can be established as expected. The following figure illustrates the end result of this test.

Figure 17. Removed External Network



To guarantee that network connections are directed over the correct interfaces on the vCloud Networking and Security Edge Gateway, the only "fail safe" option is to remove the Internet external network forcing all network traffic over the interface connected to the Internet_DR network.

Note: Upon removing an external network from an Edge Gateway, any rules associated with it are deleted. In the case of this scenario all predefined SNAT/DNAT rules were deleted.

4.5.3.2. Test 2 – Disable vCloud Networking and Security Edge Gateway vNICs

The purpose of this test is to investigate alternative methods to make sure that network traffic is routed over a specific interface on an Edge Gateway. The approach validates that disconnecting a virtual adapter has the effect of taking the interface down. This prevents network traffic from being passed to the incorrect interface. The following is the high level test procedure.

To verify lack of network connectivity

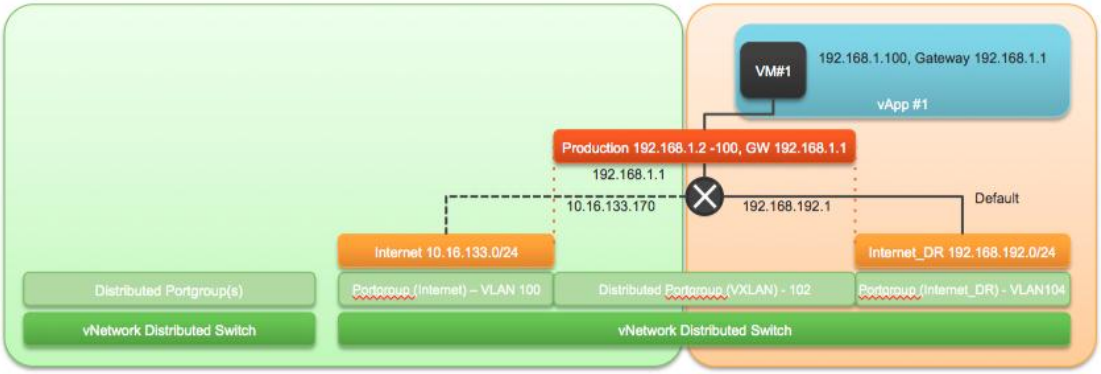
1. Attempt to connect to vCenter Server, navigate the inventory and edit the Connected status of the virtual machine settings. In vCenter Server the option to edit the configuration of the vCloud Networking and Security Edge (Edge) appliance is disabled.
2. Repeat Step 1, but from the ESXi host on which the Edge device is running. In ESXi the option to edit the configuration of the Edge appliance is enabled
3. Repeat the network connectivity steps defined in Test 1, Step 4 – Despite disabling the network adapter the Edge device appears to continue to try to route packets over the original interface.

Although the ability to update the Edge device settings is disabled within vCenter Server, it is still accessible by connecting directly to the ESXi host on which the appliance is running. The key observation during this testing is that despite disabling the virtual adapter for the Edge appliance, the guest OS of the Edge device does not correctly detecting that the interface was down. The hardware change is detected as "protocol down, device up."

4.5.3.3. Test 3 – Prove Connectivity and Verify NAT Configuration Including Failover

This test is a repeat of Test 1, but both the vCloud Networking and Security Edge Gateway (Edge Gateway) and vApp are made to fail and are then brought online at the recovery site. The following figure illustrates the end result of this test.

Figure 18. Test 3 End Result



The result of this test is consistent with the Test 1 result, even with the added changes of failing over the vApp and Edge Gateway device.

Testing and validation during the production of this example results in a solution that can help simplify the deployment of vCloud Director DR solutions in the absence of stretched Layer 2 networking. Furthermore, this solution is complementary to the solution already defined in the *VMware vCloud Director Infrastructure Resiliency Case Study* (http://www.vmware.com/files/pdf/techpaper/vcloud-director-infrastructure-resiliency.pdf) and can be implemented with relatively few additions to the existing vCloud DR recovery process.

4.5.4 Updated vCloud DR Recovery Process

Following the successful recovery of a vCloud Director management cluster, some additional steps must be included in the recovery of resource clusters to facilitate the recovery of vCloud Networking and Security Edge Gateway (Edge Gateway) appliances and vApps. The following is the high-level procedure.

To facilitate the recovery of Edge Gateway appliances and vApps

1. Restart all of the virtual machines (Edge devices) in the systems folders, one at a time.
2. For each vCloud Networking and Security Manager, retrieve the current configurations and apply the site-specific networking mapping updates.
3. Remove the primary interface from the Edge devices to allow all traffic to flow through the recovery interface.
4. Bring up the virtual machines protected by the Edge devices.

   Note: You could consider using the metadata property of objects to define site-specific configuration information that can be applied during the recovery process. The use of metadata is discussed in the automation examples of the *VMware vCloud Director Infrastructure Resiliency Case Study*.

The following table provides a high-level overview of the existing vCloud DR recovery process and an updated vCloud DR process that incorporates the solution described in this document.

Table 12. Existing versus Revised vCloud DR Process

Existing vCloud DR Process	Updated vCloud DR Process
1. Mount replicated VMFS volumes. 2. Bring recovery ESXi hosts online. 3. Power on vCloud Director workload virtual machines.	1. Mount replicated VMFS volumes. 2. Bring recovery ESXi hosts online. 3. Bring Edge Gateway device online. a. Power on affected Edge Gateway devices. b. Enable predefined services configurations for recovery site. c. Remove interface connected to production site. 4. Power on vCloud Director workload virtual machines.

4.5.4.1. Updated vCloud DR Recovery Process – API Example

In keeping with the existing vCloud DR solution there is a requirement to consider automation. While this solution offers additional simplicity and reduced configuration it is still necessary to update the configuration of multiple vCloud Networking and Security Edge Gateway (Edge Gateway) devices, which in turn can have multiple NAT or firewall rules defined. In this example, the approach to automating steps 3.b and 3.c is considered. Step 3.a is excluded because this is only a case of identifying vCloud Networking and Security Edge devices (easily identified by their location in the system virtual datacenter resource pools) and issuing a Power On request. (This is covered previously in the automation examples for the existing vCloud DR solution.)

In vCloud Director 1.x, the network services such as firewall, static routing, DHCP, and so on, were all associated with the organization network. However, in vCloud Director 5.1 all network services are associated with the Edge Gateway instead of the organization virtual datacenter network.

4.5.4.2. Enable predefined services configurations for recovery site

The following high-level example procedure uses the vCloud Director API to get information about the vCloud Networking and Security Edge Gateway (Edge Gateway) devices, modify that information, and update the device configuration.

To use the vCloud Director API to get and modify information for the Edge Gateway devices

1. Authenticate to vCloud Director (Section 0).
2. Get and return the Edge Gateway devices (Section 4.5.4.5).
3. Get and return the specific Edge Gateway device current configuration (Section 4.5.4.6).
4. Modify the XML to reflect the new configuration (Section 0).
5. Update the Edge Gateway device configuration (Section 4.5.4.8).

For illustration, the example updates a given Edge Gateway device to change the HA status. In a full recovery scenario, all configuration elements can be updated by editing or adding the correct section in the XML document that represents the Edge Gateway service configuration.

After you have successfully implemented your vCloud DR solution, you can look at how to use and implement this solution using the vCloud API. This section of the document introduces you to the VMware vCloud™ API and, in particular, the Edge Gateway API and Query Service API.

The vCloud API uses HTTP requests (which are often executed by a script or other higher-level language) as a way of making what are essentially remote procedure calls that create, modify, or delete the objects defined by the API. This vCloud REST API is defined by a collection of XML documents that represent the objects on which the API operates. The operations themselves (HTTP requests) are generic to all HTTP clients.

The vCloud REST API work flows fall into a pattern that includes only two fundamental operations:

Make an HTTP request (typically GET, PUT, POST, or DELETE). The target of this request is either a well-known URL (such as the vCloud Director URL) or a link obtained from the response to a previous request.

Examine the response, which can be an XML document or an HTTP response code.

- If the response is an XML document, it can contain links or other information about the state of an object.
- If the response is an HTTP response code, it indicates whether the request succeeded or failed, and can be accompanied by a URL that points to a location from which additional information can be retrieved.

4.5.4.3. Using cURL

Using tools such as cURL, we can consume the vCloud Networking and Security REST API. There is no need for document descriptions, because only touching each URL with the appropriate method and data causes an immediate response.

cURL, sometimes written as *curl*, is a set of C-based libraries in PHP that supports HTTP GET. cURL supports the following command line options:

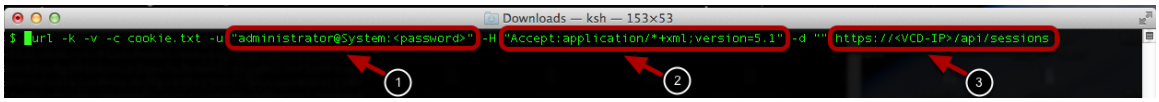
- -i (HTTP) Include the HTTP-header in the output. The HTTP-header includes things like server-name, date of the document, and the HTTP-version.
- -k Allow connections to SSL sites without certificates.
- -H Specify a custom HTTP header to pass to the server.
- -X Specifies a custom request method to use when communicating with the HTTP server. The specified request is used instead of the method otherwise used (which defaults to GET). Read the HTTP 1.1 specification for details and explanations. Common additional HTTP requests include POST and DELETE.

4.5.4.4. Authenticate to vCloud Director

The following example shows how to authenticate to vCloud Director.

Request

POST https://vcloud.cloudlab.com/api/sessions




```
$ curl -k -v -c cookie.txt -u "administrator@System:akimbi" -H
"Accept:application/*+xml;version=5.1" -d "" https://<VCD-IP>/api/sessions
```

Response

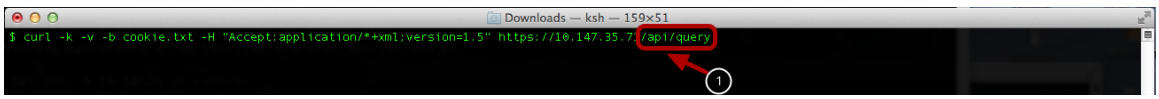
```
* About to connect() to <VCD-IP> port 443 (#0)
*   Trying <VCD-IP>... connected
* Connected to <VCD-IP> (<VCD-IP>) port 443 (#0)
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using DHE-RSA-AES256-SHA
* Server certificate:
*   subject: C=US; ST=California; L=Palo Alto; O=VMware, Inc.; CN=*.eng.vmware.com
*   start date: 2009-11-17 00:00:00 GMT
*   expire date: 2012-11-20 23:59:59 GMT
*   common name: *.example.vmware.com (does not match '<VCD-IP>')
*   issuer: C=US; O=DigiCert Inc; OU=www.digicert.com; CN=DigiCert High Assurance
CA-3
*   SSL certificate verify ok.
* Server auth using Basic with user 'administrator@System'
> POST /api/sessions HTTP/1.1
> Authorization: Basic YWRtaW5pc3RyYXRvckBTeXN0ZW06YWtpbWJp
> User-Agent: curl/7.21.4 (universal-apple-darwin11.0) libcurl/7.21.4
OpenSSL/0.9.8r zlib/1.2.5
> Host: <VCD-IP>
> Accept:application/*+xml;version=5.1
> Content-Length: 0
> Content-Type: application/x-www-form-urlencoded
>
```

```
< HTTP/1.1 200 OK

< Date: Tue, 24 Jul 2012 18:11:38 GMT

< x-vcloud-authorization: +UDXmIeKSZ9QnPpg9OPNEhtC5QgTUzvNmyJ6IZgx6hI=

* Added cookie vcloud-token="+UDXmIeKSZ9QnPpg9OPNEhtC5QgTUzvNmyJ6IZgx6hI=" for
domain <VCD-IP>, path /, expire 0

< Set-Cookie: vcloud-token=+UDXmIeKSZ9QnPpg9OPNEhtC5QgTUzvNmyJ6IZgx6hI=; Secure;
Path=/

< Content-Type: application/vnd.vmware.vcloud.session+xml;version=5.1

< Date: Tue, 24 Jul 2012 18:11:39 GMT

< Content-Length: 1259

<

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

<Session xmlns="http://www.vmware.com/vcloud/v1.5" user="administrator"
org="System" type="application/vnd.vmware.vcloud.session+xml"
href="https://10.147.50.34/api/session/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5 http://<VCD-IP>
/api/v1.5/schema/master.xsd">

    <Link rel="down" type="application/vnd.vmware.vcloud.orgList+xml"
href="https://<VCD-IP> /api/org/"/>

    <Link rel="down" type="application/vnd.vmware.admin.vcloud+xml"
href="https://<VCD-IP> /api/admin/"/>

    <Link rel="down" type="application/vnd.vmware.admin.vmwExtension+xml"
href="https://<VCD-IP>/api/admin/extension"/>

    <Link rel="down" type="application/vnd.vmware.vcloud.org+xml" name="System"
href="https://<VCD-IP>/api/org/a93c9db9-7471-3192-8d09-a8f7eeda85f9"/>

    <Link rel="down" type="application/vnd.vmware.vcloud.query.queryList+xml"
href="https://<VCD-IP>/api/query"/>

    <Link rel="entityResolver" type="application/vnd.vmware.vcloud.entity+xml"
href="https://<VCD-IP>/api/entity/"/>

    <Link rel="down:extensibility"
type="application/vnd.vmware.vcloud.apiextensibility+xml" href="https://<VCD-
IP>/api/extensibility"/>

</Session>

* Connection #0 to host <VCD-IP> left intact

* Closing connection #0

* SSLv3, TLS alert, Client hello (1):
```

(Optional) Return the vCloud Director metadata.

Request

POST https://vcloud.cloudlab.com/api/query




```
$ curl -k -v -b cookie.txt -H "Accept:application/*+xml;version=5.1" https://<VCD-
IP>/api/query
```

This presents a list if many elements, such as organization, adminOrgNetwork, providerVdc, externalNetwork, and edgeGateway.

Response (Modified to separate the vCloud Director objects.)

Organization*:*

```
<Link rel="down" type="application/vnd.vmware.vcloud.query.records+xml"
name="organization" href="https://<VCD-
IP>/api/query?type=organization&amp;format=records"/>
```

adminOrgNetwork:

```
<Link rel="down" type="application/vnd.vmware.vcloud.query.references+xml"
name="adminOrgNetwork" href="https://<VCD-
IP>/api/query?type=adminOrgNetwork&amp;format=references"/>
```

providerVdc:

```
<Link rel="down" type="application/vnd.vmware.vcloud.query.references+xml"
name="providerVdc" href="https://<VCD-
IP>/api/query?type=providerVdc&amp;format=references"/>
```

externalNetwork:

```
<Link rel="down" type="application/vnd.vmware.vcloud.query.references+xml"
name="externalNetwork" href="https://<VCD-
IP>/api/query?type=externalNetwork&amp;format=references"/>
```

edgeGateway**:**

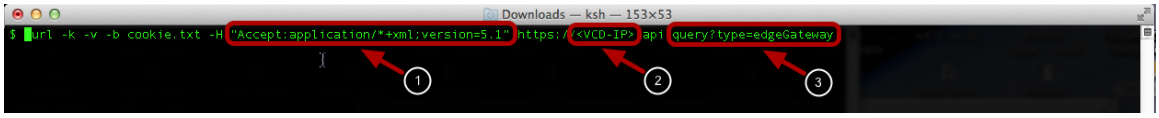
```
<Link rel="down" type="application/vnd.vmware.vcloud.query.references+xml"
name="edgeGateway" href="https://<VCD-
IP>/api/query?type=edgeGateway&amp;format=references"/>
```

4.5.4.5. Get and Return the vCloud Networking and Security Edge Gateways

The following example shows how to get information about the vCloud Networking and Security Edge Gateway (Edge Gateway).

Request

GET https://vcloud.cloudlab.com/api/query?type=edgeGateway




```
$ curl -k -v -b cookie.txt -H "Accept:application/*+xml;version=5.1" https://<VCD-
IP>/api/query?type=edgeGateway
```

Response

The following is a condensed format to show one Edge Gateway for this example, in this case **Edge-Gateway-01.**

```
<EdgeGatewayRecord vdc="https://<VCD-IP>/api/vdc/1d7f9e91-ef16-48ad-bae8-
299bfe56a54c" numberOfOrgNetworks="1" numberOfExtNetworks="1" name="Edge-Gateway-
01" isBusy="false" haStatus="UP" gatewayStatus="READY" href="https://<VCD-
IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-bdd688a64963"
isSyslogServerSettingInSync="true" taskStatus="success"
taskOperation="networkEdgeGatewayCreate" task="https://<VCD-IP>/api/task/62928cf9-
937b-4f06-ba55-01f032a32ace" taskDetails=" "/>
```

4.5.4.6. Get and Return a Specific vCloud Networking and Security Edge Gateway

The following example shows how to get information about a specific vCloud Networking and Security Edge Gateway.

Request

GET https://<VCD-IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-bdd688a64963

```
$ curl -k -v -b cookie.txt -H "Accept:application/*+xml;version=5.1"
https://10.147.50.33/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-bdd688a64963
```

Response

```
<EdgeGateway xmlns="http://www.vmware.com/vcloud/v1.5" status="1" name="Edge-
Gateway-01" id="urn:vcloud:gateway:0cf71e84-fdf6-4fa0-ae85-bdd688a64963"
type="application/vnd.vmware.admin.edgeGateway+xml" href="https://<VCD-
IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-bdd688a64963"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5
http://10.147.50.33/api/v1.5/schema/master.xsd">

    <Link rel="edit" type="application/vnd.vmware.admin.edgeGateway+xml"
href="https://<VCD-IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-
bdd688a64963"/>

    <Link rel="remove" href="https://<VCD-IP>/api/admin/edgeGateway/0cf71e84-fdf6-
4fa0-ae85-bdd688a64963"/>

    <Link rel="up" type="application/vnd.vmware.admin.vdc+xml" href="https://<VCD-
IP>/api/admin/vdc/1d7f9e91-ef16-48ad-bae8-299bfe56a54c"/>

    <Link rel="edgeGateway:redeploy" href="https://<VCD-
IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-bdd688a64963/action/redeploy"/>

    <Link rel="edgeGateway:configureServices"
type="application/vnd.vmware.admin.edgeGatewayServiceConfiguration+xml"
href="https://<VCD-IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-
bdd688a64963/action/configureServices"/>

    <Link rel="edgeGateway:reapplyServices" href="https://<VCD-
IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-
bdd688a64963/action/reapplyServices"/>

    <Link rel="edgeGateway:syncSyslogSettings" href="https://<VCD-
IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-
bdd688a64963/action/syncSyslogServerSettings"/>

    <Link rel="edgeGateway:upgrade" href="https://<VCD-
IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-
bdd688a64963/action/upgradeConfig"/>

    <Description/>

    <Configuration>

        <GatewayBackingConfig>compact</GatewayBackingConfig>

        <GatewayInterfaces>

            <GatewayInterface>

                <Name>TestBed-VC1</Name>

                <DisplayName>TestBed-VC1</DisplayName>

                <Network type="application/vnd.vmware.admin.network+xml"
name="TestBed-VC1" href="https://<VCD-IP>/api/admin/network/3ddab120-7d66-40d3-
9536-af94f23e1361"/>

                <InterfaceType>uplink</InterfaceType>

                <SubnetParticipation>

                    <Gateway>192.168.1.1</Gateway>

                    <Netmask>255.255.255.0</Netmask>

                    <IpAddress>192.168.1.8</IpAddress>

                </SubnetParticipation>
```

```
                <SubnetParticipation>
                    <Gateway>192.168.2.1</Gateway>
                    <Netmask>255.255.255.0</Netmask>
                    <IpAddress>192.168.2.7</IpAddress>
                </SubnetParticipation>
                <SubnetParticipation>
                    <Gateway>198.125.2.6</Gateway>
                    <Netmask>255.255.0.0</Netmask>
                    <IpAddress>198.125.2.12</IpAddress>
                </SubnetParticipation>
                <SubnetParticipation>
                    <Gateway>10.147.80.253</Gateway>
                    <Netmask>255.255.255.0</Netmask>
                    <IpAddress>10.147.80.217</IpAddress>
                </SubnetParticipation>
                <ApplyRateLimit>false</ApplyRateLimit>
                <InRateLimit>100.0</InRateLimit>
                <OutRateLimit>100.0</OutRateLimit>
                <UseForDefaultRoute>true</UseForDefaultRoute>
            </GatewayInterface>
            <GatewayInterface>
                <Name>MAH-VDC-Network</Name>
                <DisplayName>MAH-VDC-Network</DisplayName>
                <Network type="application/vnd.vmware.admin.network+xml" name="MAH-
VDC-Network" href="https://<VCD-IP>/api/admin/network/2d6b1a79-a249-4ba3-b863-
e3649661801f"/>
                <InterfaceType>internal</InterfaceType>
                <SubnetParticipation>
                    <Gateway>192.176.100.1</Gateway>
                    <Netmask>255.255.255.0</Netmask>
                    <IpAddress>192.176.100.1</IpAddress>
                </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>true</UseDefaultRouteForDnsRelay>
    </Configuration>
</EdgeGateway>
```

4.5.4.7. Modify the XML to Reflect the New Configuration

The entire body from the GET response in step 3 in Section 4.5.4.1 is used to make the required changes. This example shows the change of <HaEnabled> to true.

To automate the reconfiguration of the vCloud Networking and Security Edge Gateway devices

1. Create a file and copy the contents of the <EdgeGateway> ... </EdgeGateway> into this file (for example, EdgeGateway.xml).
2. Change <HaEnabled>false</HaEnabled> to <HaEnabled>true</HaEnabled>.
3. Copy and paste the contents of this file into the http PUT or cURL command in step 5.

4.5.4.8. Update the vCloud Networking and Security Edge Gateway Device Configuration

After making the preceding changes, you can update the vCloud Networking and Security Edge Gateway device using the vCloud API as in the following example.

Request

PUT https://<VCD-IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-bdd688a64963 (this is the UUID 0cf71e84-fdf6-4fa0-ae85-bdd688a64963)

```
$ curl -k -v -b cookie.txt -H "Accept:application/*+xml;version=5.1" -X
PUT --header "Content-Type:application/vnd.vmware.admin.edgeGateway+xml" -
-data @EdgeGateway.xml https://<VCD-IP>/api/admin/edgeGateway/0cf71e84-
fdf6-4fa0-ae85-bdd688a64963
```

Response

```
* About to connect() to <VCD-IP> port 443 (#0)
*   Trying <VCD-IP>... connected
* Connected to <VCD-IP> (<VCD-IP>) port 443 (#0)
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server key exchange (12):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using DHE-RSA-AES256-SHA
* Server certificate:
*   subject: C=US; ST=California; L=Palo Alto; O=VMware, Inc.; CN=*.eng.vmware.com
```

```
*   start date: 2009-11-17 00:00:00 GMT
*   expire date: 2012-11-20 23:59:59 GMT
*   common name: *.eng.vmware.com (does not match <VCD-IP>)
*   issuer: C=US; O=DigiCert Inc; OU=www.digicert.com; CN=DigiCert High Assurance
CA-3
*   SSL certificate verify ok.
> PUT /api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-bdd688a64963 HTTP/1.1
> User-Agent: curl/7.21.4 (universal-apple-darwin11.0) libcurl/7.21.4
OpenSSL/0.9.8r zlib/1.2.5
> Host: <VCD-IP>
> Cookie: vcloud-token=1hi8kZ4tNOnSnv3aq6/gSrDHlTPyYrBXQ5a2CdmX8C4=
> Accept:application/*+xml;version=5.1
> Content-Type:application/vnd.vmware.admin.edgeGateway+xml
> Content-Length: 4631
> Expect: 100-continue
>
< HTTP/1.1 100 Continue
< HTTP/1.1 202 Accepted
< Date: Fri, 24 Jul 2012 10:08:13 GMT
< Date: Fri, 24 Jul 2012 10:08:15 GMT
< Location: https://<VCD-IP>/api/task/e0c73c28-2d5b-4e9d-a304-bc6b3667f18a
< Content-Type: application/vnd.vmware.vcloud.task+xml;version=5.1
< Content-Length: 1331
<
<?xml version="1.0" encoding="UTF-8"?>
<Task xmlns="http://www.vmware.com/vcloud/v1.5" status="running" startTime="2012-
07-20T03:08:15.571-07:00" serviceNamespace="com.vmware.vcloud"
operationName="edgeGatewayUpdate" operation="Updating EdgeGateway (0cf71e84-fdf6-
4fa0-ae85-bdd688a64963)" expiryTime="2012-10-18T03:08:15.571-07:00"
cancelRequested="false" name="task" id="urn:vcloud:task:e0c73c28-2d5b-4e9d-a304-
bc6b3667f18a" type="application/vnd.vmware.vcloud.task+xml" href="https://<VCD-
IP>/api/task/e0c73c28-2d5b-4e9d-a304-bc6b3667f18a"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5 http://<VCD-
IP>/api/v1.5/schema/master.xsd">
    <Link rel="task:cancel" href="https://<VCD-IP>/api/task/e0c73c28-2d5b-4e9d-
a304-bc6b3667f18a/action/cancel"/>
    <Owner type="application/vnd.vmware.admin.edgeGateway+xml" name=""
href="https://<VCD-IP>/api/admin/edgeGateway/0cf71e84-fdf6-4fa0-ae85-
bdd688a64963"/>
    <User type="application/vnd.vmware.admin.user+xml" name="system"
href="https://<VCD-IP>/api/admin/user/55c1d771-b2e2-4255-8387-7f6da1e0e3f1"/>
    <Organization type="application/vnd.vmware.vcloud.org+xml" name="MAH"
href="https://<VCD-IP>/api/org/60b44eb5-0e98-45bc-b96b-25549ce03033"/>
    <Progress>0</Progress>
    <Details/>
</Task>
* Connection #0 to host <VCD-IP> left intact
```

```
* Closing connection #0
```

4.5.5 Design Considerations

- The vCloud Networking and Security Edge (Edge) devices can be available in only one site at a time. Having the hosts in maintenance mode on the recovery side enforces this and also keeps them in sync with all of the changes that happen at the primary site.
- Removal of the primary interface after failover avoids loss of traffic for the primary site to a nonexistent interface in the case of partial environment recovery. This is because Edge always prefers the locally attached interface for sending traffic instead of sending all traffic out of the default interface. The designation of a default interface is used only for cases where there are no locally attached networks. The Edge device uses this interface to send all unknown destination traffic to its designated default router. It is prudent in the case where the environment might be failing back to its primary site to save all interface-based rules and configurations before removal. This is because all configurations associated with the interface are removed as soon as an interface is removed.
- vCloud primary sites that have direct organization networks do not use VXLAN so the recovery process is identical to the existing vCloud DR recovery process. All of the vApps on that network must have IP addresses reassigned to the correct addressing used in the recovery site Layer 3 network.
- vCloud primary sites that use isolated networks that are VLAN-backed must be recreated at the recovery site using the associated VLAN IDs available at the recovery site and the vApps reconnected to the new network. If the Isolated networks were port group-backed, the port groups still exist in the recovery site, but their definitions must be revisited to verify that their configurations remain valid.
- vCloud primary sites that use routed or isolated VXLAN-backed networks are easy to recover due to VXLAN technology.

4.5.6 References

- *vCloud Director Infrastructure Resiliency Case Study*
  www.vmware.com/files/pdf/techpaper/vcloud-director-infrastructure-resiliency.pdf
- *vCloud Director API Programming Guide*
  http://www.vmware.com/pdf/vcd_15_api_guide.pdf

4.6 VCDNI-Backed Organization Network

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

4.6.1 Background

A network pool is an object in vCloud that creates and manages isolated networks using vSphere port groups. The vSphere port groups can be pre-provisioned in vSphere or created dynamically by vCloud as needed for organization or vApp networks.

The following types of network pools are available in vCloud Director:

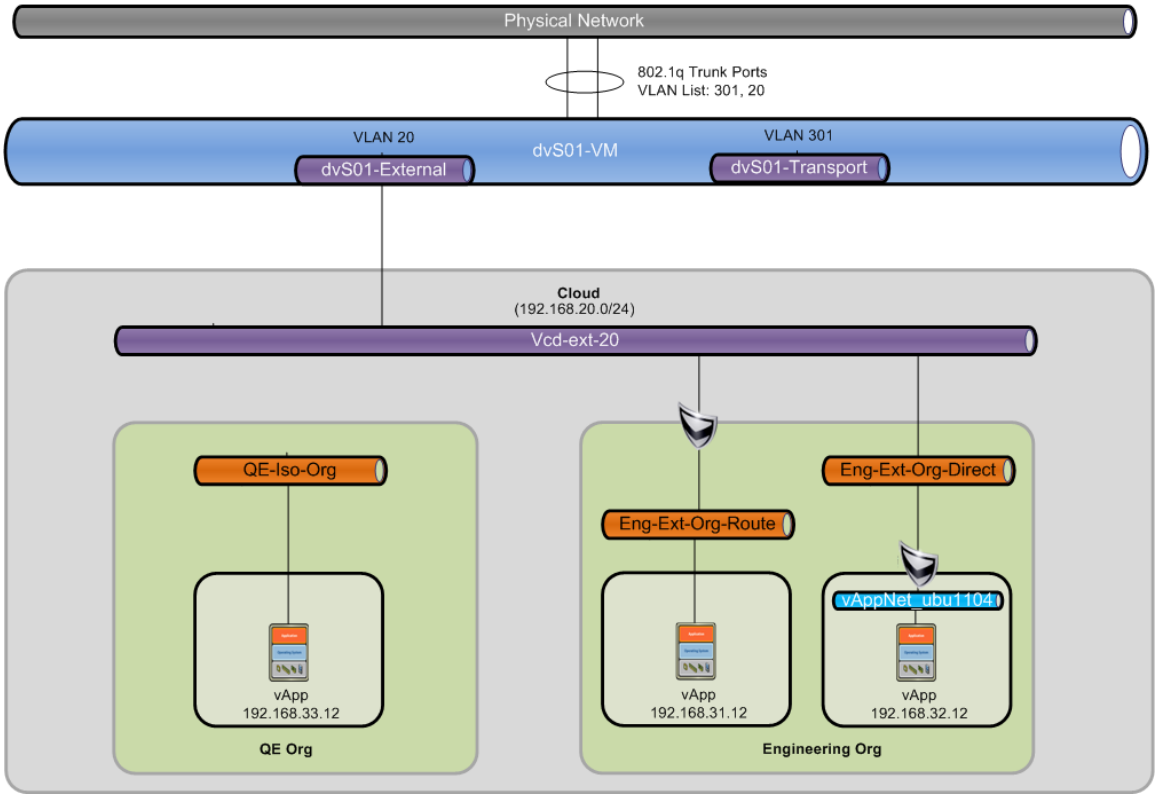
- VXLAN (vSphere port groups dynamically created)
- VLAN-backed (vSphere port groups dynamically created)
- vCloud Network Isolation-backed (vSphere port groups dynamically created)
- vSphere port group-backed (vSphere port groups manually created)

4.6.2 Example

This example documents the VCDNI-backed network pool.

The VCDNI-backed network pool example demonstrates how VCDNI networks are created automatically in vSphere and used in vCloud Director.

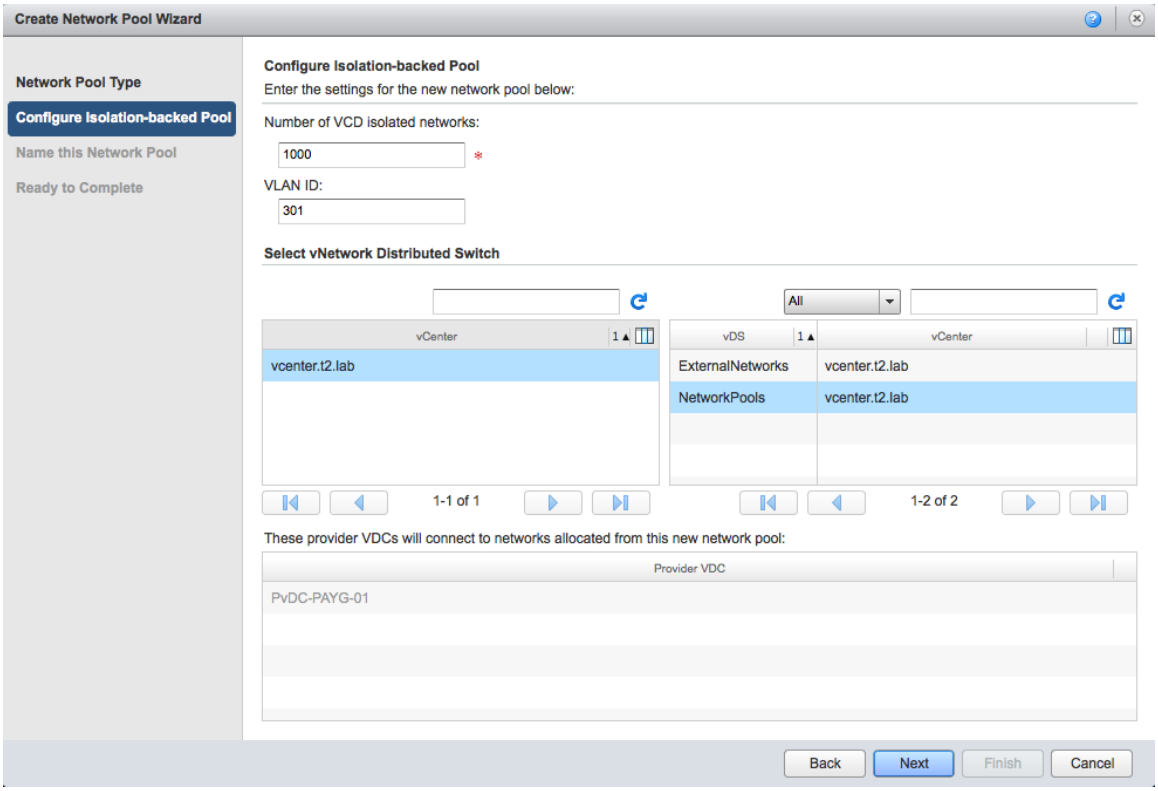
Figure 19. VCDNI Network Pool Example Configuration



The prerequisites are as follows:

- vSphere Administrator – A virtual distributed vSwitch that is connected to all vSphere (ESXi) hosts that are in the cluster for the underlying provider virtual datacenter.
- Network Administrator – All physical switch port uplinks to the distributed vSwitch configured as 802.1Q VLAN Trunk Ports and configured to allow VLAN 20 and VLAN 301.
- Cloud Administrator – vCloud Director VCDNI-backed network pool created with VLAN ID 301.
- Cloud Administrator – (optional) A vCloud Director external network (vcd-ext-20), if external connectivity is needed.

Figure 20. VCDNI-Backed Network Pool Creation



Cloud Administrator – Two Organizations (Engineering and QE) provisioned.

- Engineering organization – “High Engineering PAYG” organization virtual datacenter
  - Network 1 – “Eng-Ext-Org-Route” organization virtual datacenter routed
- Engineering organization – “Default Engineering PAYG” Organization virtual datacenter
  - Network 2 – “Eng-Ext-Org-Direct” organization virtual datacenter direct network
    - Network 3 – “vAppNet-ubu1104” vApp network
- QE organization – “High QE PAYG” organization virtual datacenter
  - Network 4 – “QE-Iso-Org” organization virtual datacenter Isolated network

During the creation of an organization virtual datacenter, you can choose the network pool (the VCDNI pool created above) to associate with this virtual datacenter. Multiple organizations and different virtual datacenters within an organization can share the same network pool but where appropriate will be assigned separate and isolated networks from the pool.

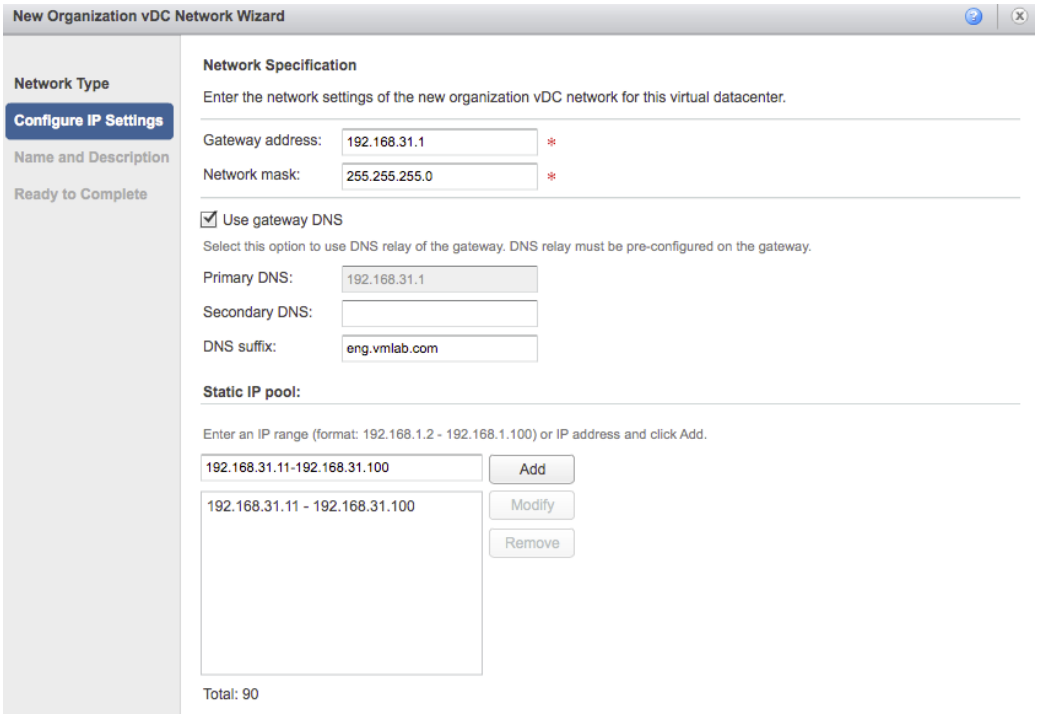
Table 13. vCloud Director Networks

Network	Network Type	Organization	Organization Virtual Datacenter	Network Pool	Subnet
Eng-Ext-Org-Route	Organization virtual datacenter routed	Engineering	High Engineering PAYG	VCDNI pool	192.168.31.0/24
Eng-Ext-Org-Direct	Organization direct	Engineering	Default Engineering PAYG	N/A	192.168.20.0/24
vAppNet_ubu1104	vApp NAT	Engineering	Default Engineering PAYG	VCDNI pool	192.168.32.0/24
QE-Iso-Org	Organization isolated	QE	High QE PAYG	VCDNI pool	192.168.33.0/24

Table 13 illustrates that only the direct connect organization virtual datacenter network does not use a network from the network pool. Organization direct connected networks use a bridged connection from the external network requiring IP configuration on the virtual machines that matches the physical network IP configuration.

After the network pool is created and associated with an organization virtual datacenter the network pools can be consumed. Whenever a routed or isolated organization virtual datacenter or vApp network is created, vCloud Director automatically provisions a port group on dvS01. These port groups are created automatically and do not share the same VLAN (301). However, each port group is treated as a separate Layer 2 networks by virtue of the VCDNI technology. The Administrator must define only the IP address settings for this network, as shown in the following figure.

Figure 21. Organization Virtual Datacenter Network – IP Address Settings



The VCDNI-backed network pool example uses the VMware VCDNI filter driver to multiplex an individual VLAN into many separate broadcast domains. The dvSwitch delivers the appropriate Ethernet frames to the appropriate port group based on the match of which vNIC belongs to which port group. For example, a broadcast from a particular vNIC will be delivered only to the other vNICs connected to the same port group even though the other port groups share the same VLAN (301), which is the transport VLAN configured for the VCDNI network pool.

This isolation persists only inside the vSphere boundaries. However it allows isolated communication between vApps connected to the same network pool even if they are on different ESXi hosts in the vSphere cluster.

4.6.3 Design Implications

- Only specific vCloud network types consume network pools, as follows:
  - Organization virtual datacenter routed networks.
  - Organization virtual datacenter isolated networks.
  - vApp networks.
- Although the VCDNI-backed network pool provides Layer 2 (Ethernet) isolation, the use of routing or NAT capabilities at the vCloud Networking and Security Edge device can provide connectivity to externally connected networks at Layer 3.
- The VCDNI uses encapsulation technology and therefore requires a larger Ethernet frame to be sent over the wire. To eliminate fragmentation of Ethernet frames, increase the MTU size on the physical network and network pool.

- It is recommended to increase the MTU size on the physical devices backing the VCDNI network pool to 1524 bytes.
- It is also recommended to increase the MTU size on the network pool itself to 1524 bytes.

See the vCloud Director documentation for more information.

4.7 VLAN ORG Network

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

4.7.1 Background

A network pool is an object in vCloud that creates and manages isolated networks using vSphere port groups. The vSphere port groups can be pre-provisioned in vSphere or created dynamically by vCloud as needed for organization or vApp networks.

The following types of network pools are available in vCloud Director:

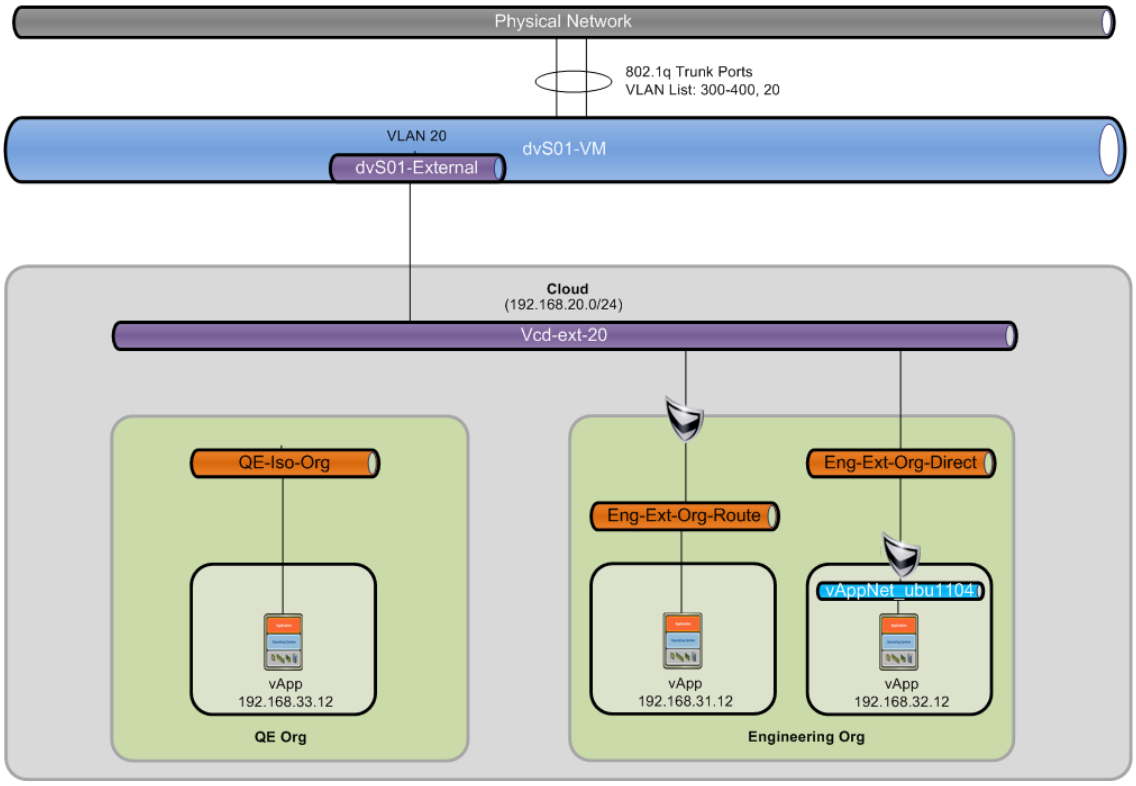
- VXLAN (vSphere port groups dynamically created).
- VLAN-backed (vSphere port groups dynamically created).
- vCloud Network Isolation-backed (vSphere port groups dynamically created).
- vSphere port group-backed (vSphere port groups manually created).

This example documents the VLAN-backed network pool.

4.7.2 Example

The VLAN-backed network pool example (Figure 22) demonstrates how the VLAN networks are created automatically in vSphere and used in vCloud Director.

Figure 22. VLAN Network Pool Example Configuration

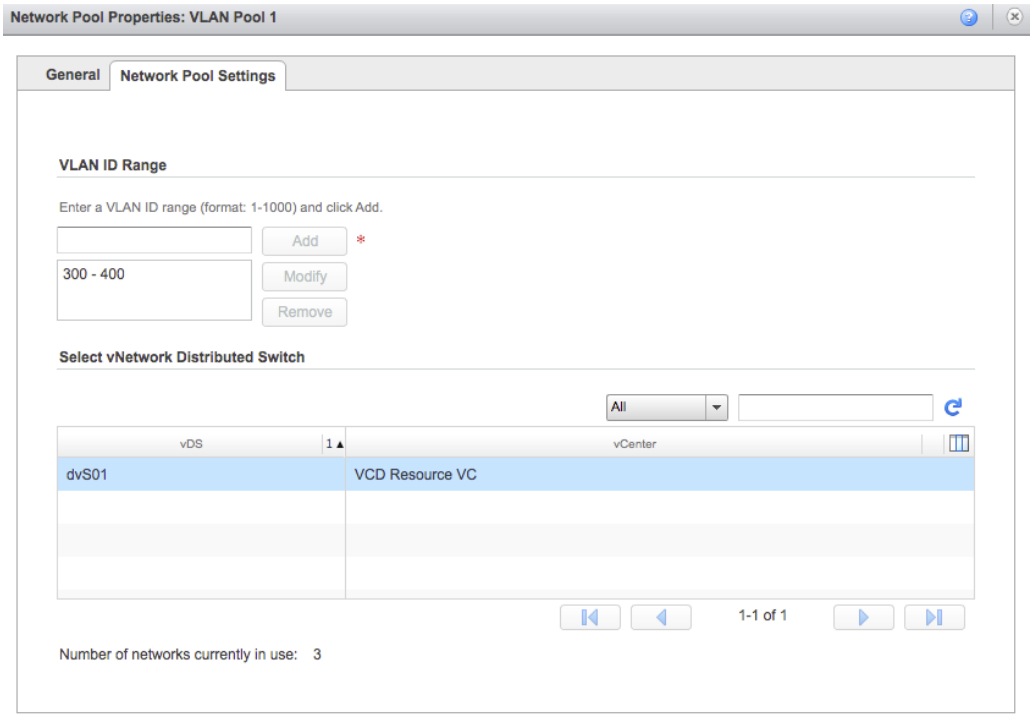


The prerequisites for this configuration are as follows:

- vSphere Administrator – A virtual distributed vSwitch that is connected to all vSphere (ESXi) hosts that are in the cluster for the underlying provider virtual datacenter.
- Network Administrator – All physical switch port uplinks to the distributed vSwitch configured as 802.1Q VLAN trunk ports and configured to allow VLANs 300–400 and VLAN 20 for the external network (vcd-ext-20).
- Cloud Administrator – vCloud Director VLAN-backed network pool created with a VLAN ID Range of 300–400.
- Cloud Administrator – (optional) A vCloud Director external network (vcd-ext-20) if external connectivity is needed.

The VLAN-backed network pool settings are shown in Figure 23.

Figure 23. VLAN-Backed Network Pool Settings



Cloud Administrator – Two Organizations (Engineering and QE) provisioned.

- Engineering organization – "High Engineering PAYG" organization virtual datacenter
  - Network 1 – "Eng-Ext-Org-Route" organization virtual datacenter routed network
- Engineering organization – "Default Engineering PAYG" organization virtual datacenter
  - Network 2 – "Eng-Ext-Org-Direct" organization virtual datacenter direct network
    - Network 3 – "vAppNet-ubu1104" vApp network
- QE organization – "High QE PAYG" organization virtual datacenter
  - Network 4 – "QE-Iso-Org" organization virtual datacenter Isolated network

During the creation of an organization virtual datacenter, you can choose the network pool (VLAN Pool 1) to associate with this virtual datacenter. Multiple and different virtual datacenters within an organization can share the same network pool, but where appropriate they are assigned separate and isolated networks from the pool.

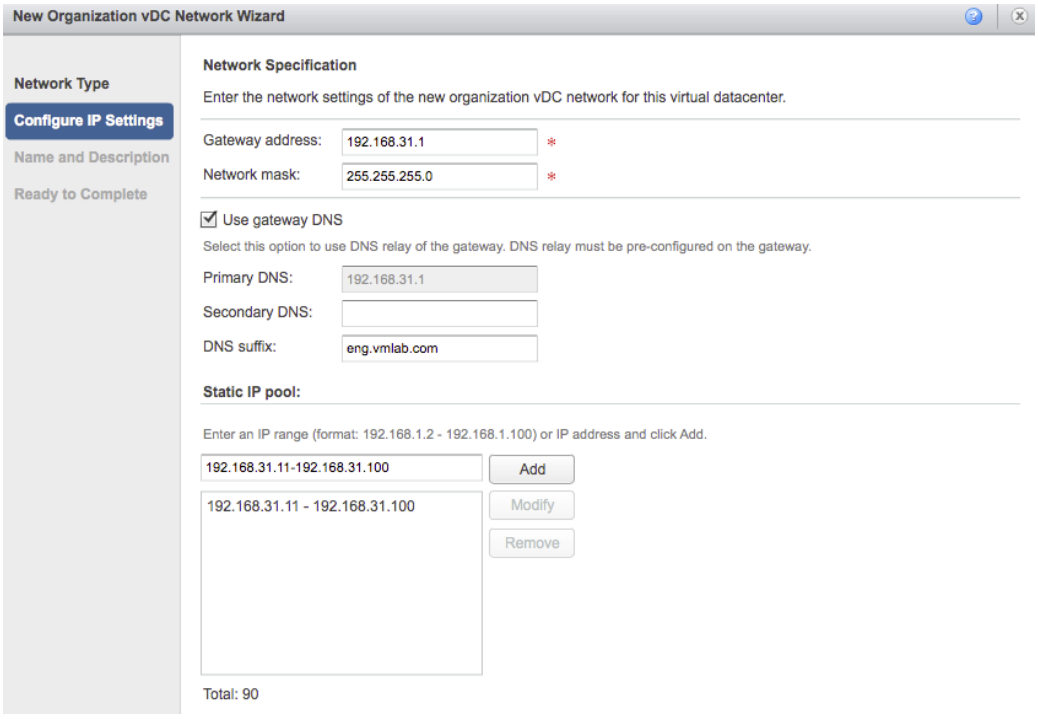
Table 14. vCloud Director Networks

Network	Network Type	Org	Organization Virtual Datacenter	Network Pool	Subnet
Eng-Ext-Org-Route	Organization virtual datacenter routed	Engineering	High Engineering PAYG	VLAN Pool 1	192.168.31.0/24
Eng-Ext-Org-Direct	Organization direct	Engineering	Default Engineering PAYG	N/A	192.168.20.0/24
vAppNet_ubu1104	vApp NAT	Engineering	Default Engineering PAYG	VLAN Pool 1	192.168.32.0/24
QE-Iso-Org	Organization isolated	QE	High QE PAYG	VLAN Pool 1	192.168.33.0/24

Table 14 illustrates that only the direct connect organization virtual datacenter network does not use a network from the network pool. Organization direct connected networks use a bridged connection from the external network requiring IP configuration on the virtual machines that matches the physical network IP configuration.

After the network pool is created and associated with an organization virtual datacenter the network pools can be consumed. Whenever a routed or isolated organization virtual datacenter or vApp network is created, vCloud Director automatically provisions a port group on dvS01 and assigns it a VLAN from the range that was defined for VLAN Pool 1 (300–400). The only thing that needs to be completed by the Administrator is to define the IP address settings for this network, as shown in Figure 24.

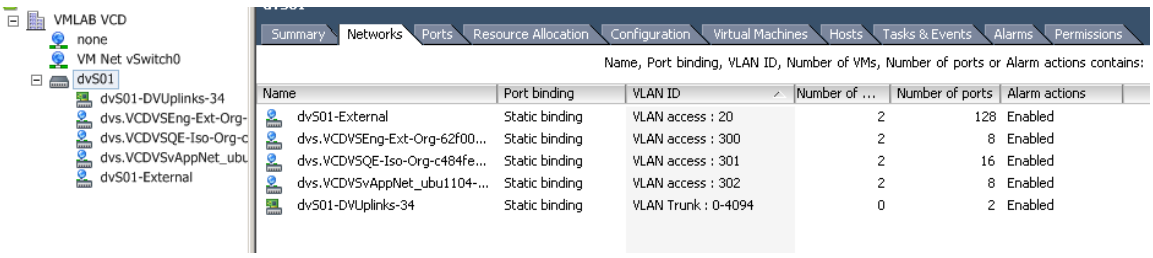
Figure 24. Organization Virtual Datacenter Network – IP Address Settings



The VLAN-backed network pool example (Figure 25) leverages 802.1Q VLAN trunk ports to allow the physical switching infrastructure to pass all VLANs configured (300–400) to the ESXi hosts, while still keeping the individual VLANs separated and in separate broadcast domains. The dvSwitch delivers the appropriate Ethernet frames to the appropriate port group based on a match of the VLAN tag on the frame and the VLAN associated with the port group. The dvSwitch port groups remove the VLAN tag from the Ethernet frame and deliver it to the appropriate virtual machine. This architecture is commonly referred to as *Virtual Switch Tagging* or VST.

This isolation also persists across the physical switching infrastructure, allowing isolated communication between virtual machines connected to the same vCloud Director network even if they are on different ESXi hosts in the vSphere cluster.

Figure 25. Network Pool Corresponding vSphere Port Groups



4.7.3 Design Implications

- Only specific vCloud network types consume network pools.
  - Organization virtual datacenter routed network.
  - Organization virtual datacenter isolated network.
  - vApp networks.
- There is a limit of 4,094 VLANs (1–4094) allowed in the 802.1Q standard.
- VLAN 0 and VLAN 4095 are reserved in the 802.1Q standard.
- Although the VLAN-backed network pool provides Layer 2 (Ethernet) isolation, the use of routing or NAT capabilities at the vCloud Networking and Security Edge instance can provide connectivity to externally connected networks at Layer 3.

5. Storage Design Examples

5.1 vApp Snapshot

Deployment Models: public.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

5.1.1 Background

In vCloud Director 5.1, vApp Users and vApp authors can now create, revert, and delete snapshots from the vCloud Director user interface or from the vCloud APIs.

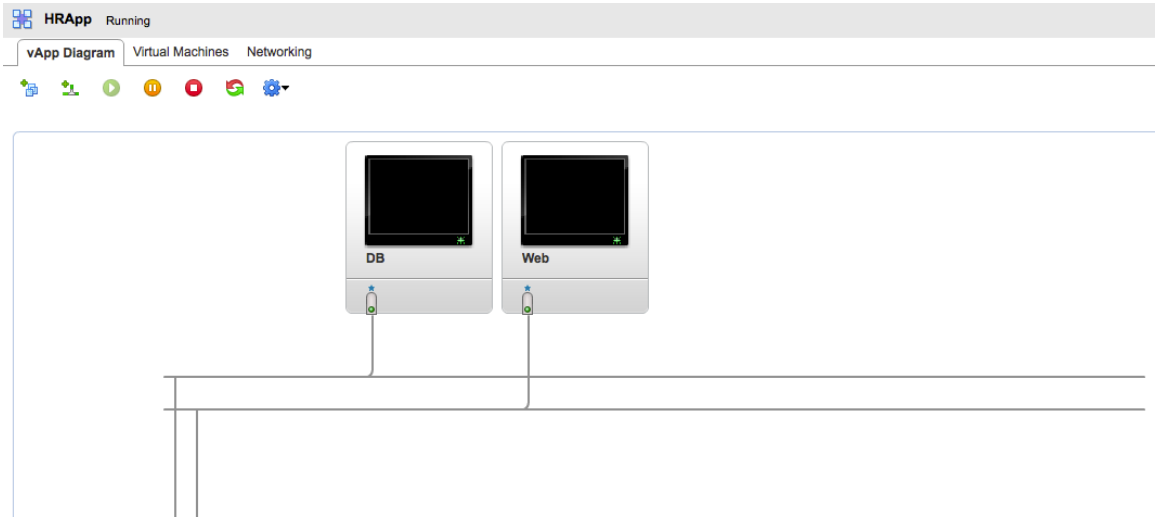
5.1.2 Use Case

This example shows how a vCloud consumer of Public Cloud resources who is responsible for a two-tier application can manipulate snapshots at both the virtual machine level and the vApp level. This is especially useful for patching and testing where a rollback might be needed. Snapshots are not intended to be used as a backup mechanism and generally should not be kept for long periods of time.

5.1.3 Example

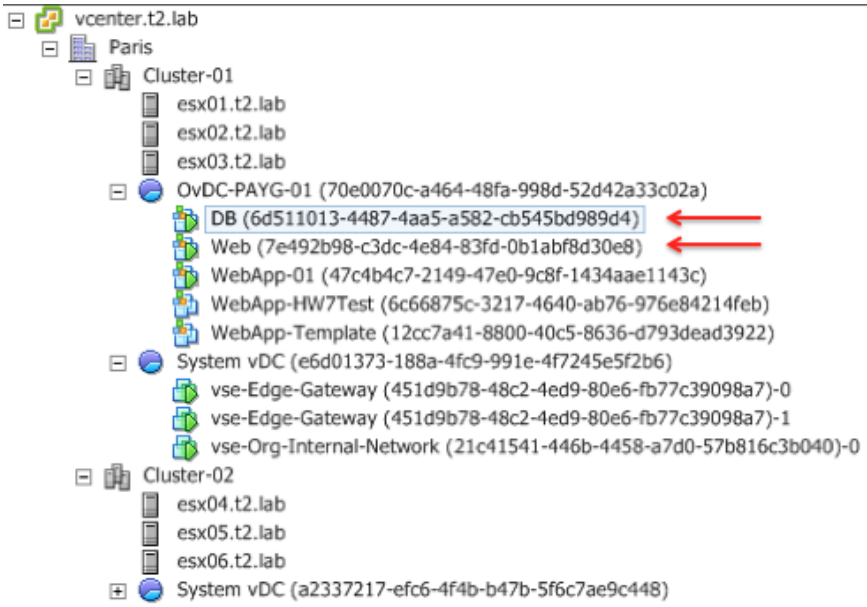
The following figure shows the basic structure of the two-tier vApp. It contains a Web virtual machine and a database virtual machine, with both running.

Figure 26. Two-Tier vApp



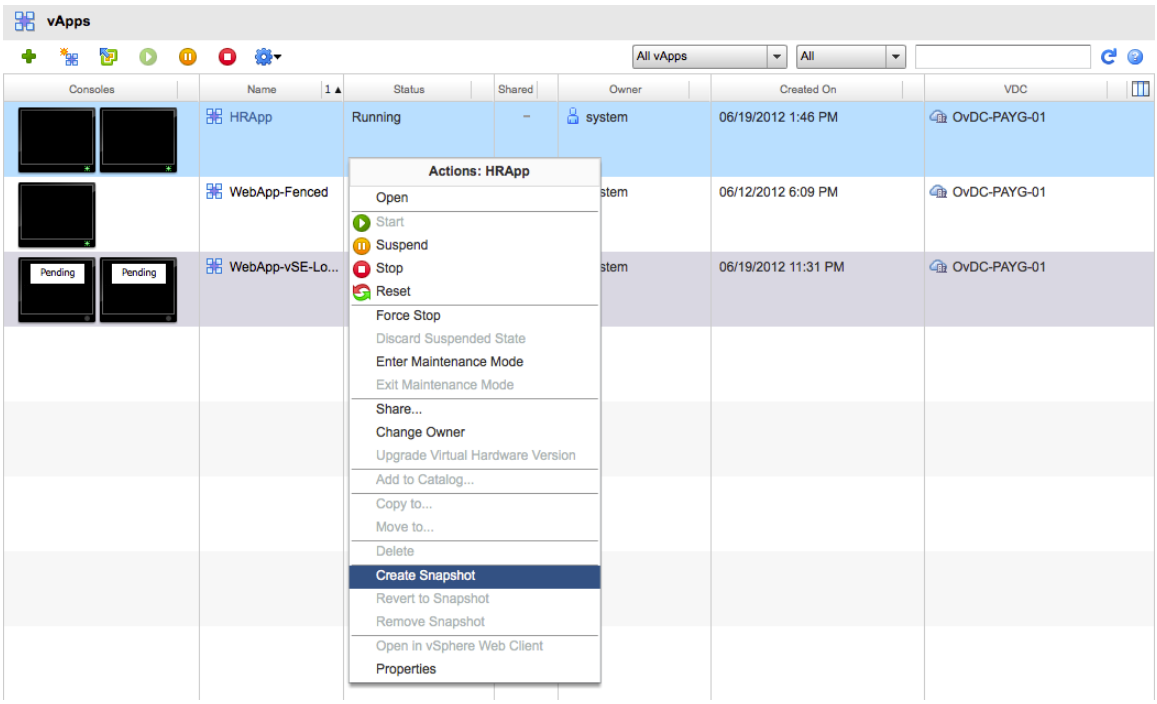
These virtual machines appear in the vSphere Client as shown in Figure 27. The vSphere Client is used only for illustration. The vCloud Director user interface is used to create, revert, and delete the snapshots.

Figure 27. Two-Tier vApp as Seen in vCenter



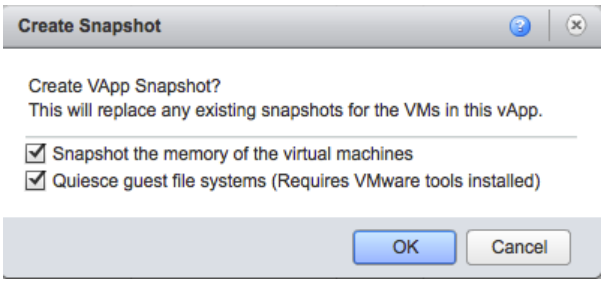
You can take a snapshot of the entire vApp while the vApp is running. Right-click the vApp and select **Create Snapshot** as shown in Figure 28. All virtual machines in the vApp have snapshots.

Figure 28. The Process of Creating a vApp Snapshot



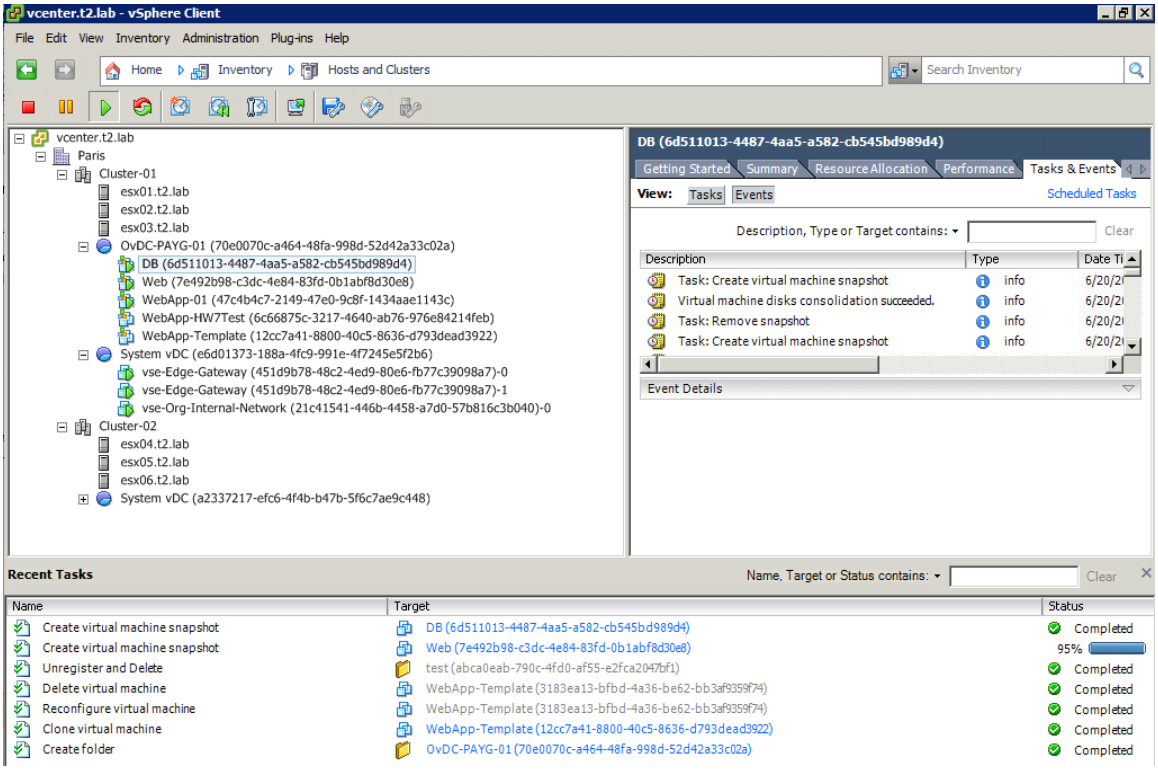
While the vApp is running, vCloud Director can take a snapshot of its memory or quiesce its file system. Quiescing the file system requires VMware Tools in both virtual machines. This is shown in Figure 29.

Figure 29. Snapshot Options



The snapshots for the two virtual machines contained in the vApp are created in vSphere. In Figure 30, the database virtual machine snapshot has just been taken and the Web virtual machine snapshot is about to complete.

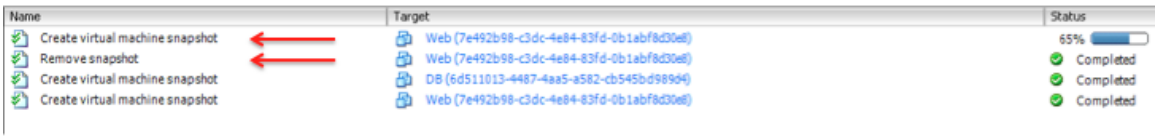
Figure 30. Snapshot Creation



The virtual machines are in snapshot mode and the consumer can start making changes to those virtual machines. If the consumer determines during testing that an additional snapshot should be taken for the Web virtual machine, the consumer can select the single virtual machine in vCloud Director, right-click, and again select **Create Snapshot**. This is illustrated in Figure 31.

Note: vCloud Director supports only a single snapshot for each virtual machine. If you create a snapshot for a virtual machine that already has an existing snapshot it deletes the existing snapshot and creates a new one. This has the effect of committing the changes from the first snapshot to the guest OS image or VMDK.

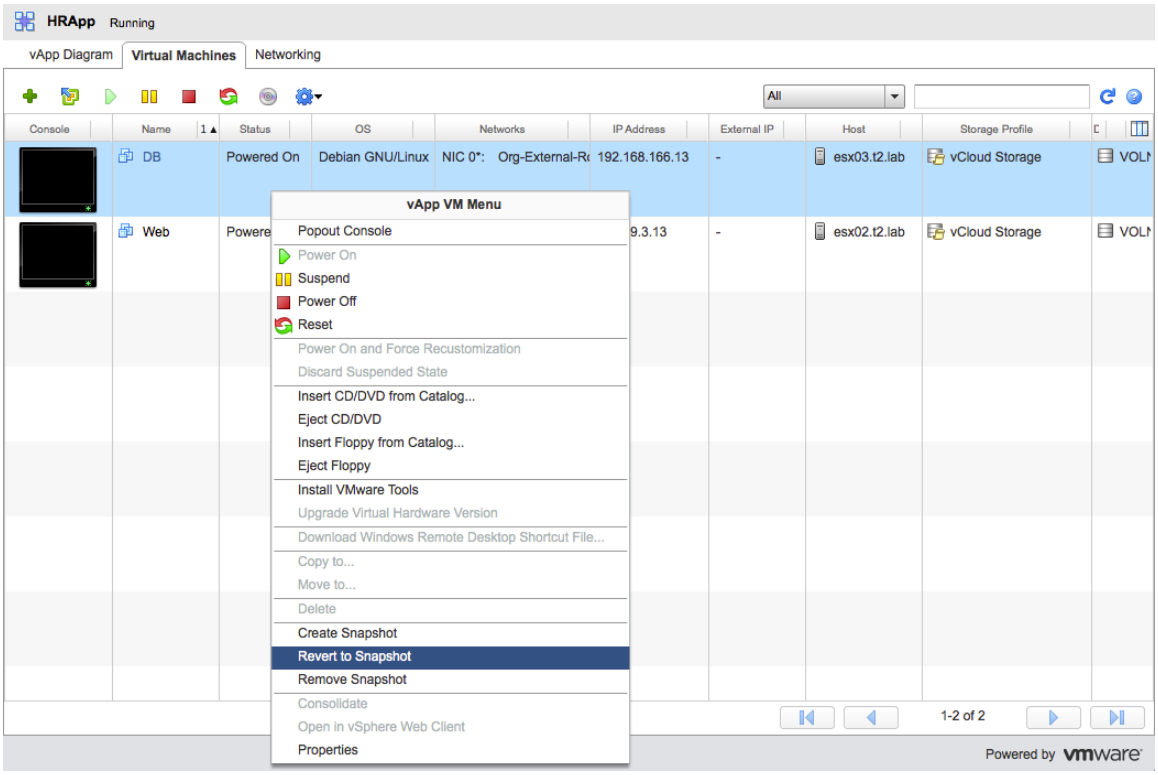
Figure 31. Creation of an Additional Snapshot for the Web Virtual Machine



After the tests are completed and the consumer determines that the Web virtual machine is working, the snapshots can be deleted. This process saves and consolidates all of the post-snapshot changes to the guest OS in the VMDK. In vCloud Director, as in vSphere, deleting a snapshot commits the changes and removes the virtual machine from snapshot mode.

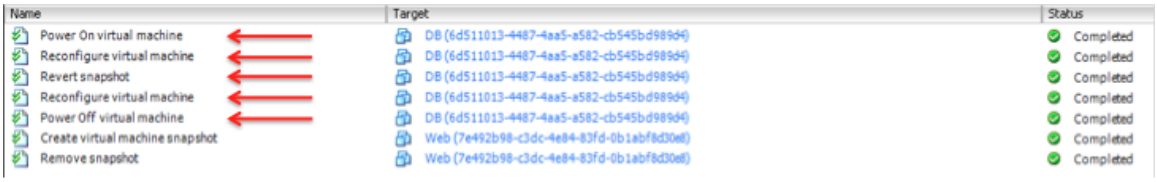
Conversely, the consumer might determine that the database virtual machine must be rolled back to its original state at the time the snapshot was created. To roll back the snapshot, the consumer right-clicks the database virtual machine and selects **Revert to Snapshot**, as shown in Figure 32. Rolling back removes the virtual machine from snapshot mode, bringing the virtual machine back to the point in time of the snapshot, and starts a new snapshot file. To completely leave snapshot mode you must delete the snapshot.

Figure 32. The Consumer Rolls Back the DB Virtual Machine



To roll back or revert a running virtual machine, vSphere must power it off, remove the snapshot, and restart the virtual machine from the original VMDK file, but with a new snapshot disk to which changes are logged. This is shown in Figure 33.

Figure 33. vSphere Tasks When Reverting a Snapshot



The consumer can issue the snapshot command at either the vApp level or the virtual machine level from within vCloud Director.

In this example, the consumer has taken a vApp level snapshot and then manipulated the individual virtual machines because each virtual machine required different operations. If a vApp level snapshot command is issued, the same command is propagated to all the virtual machines that comprise the vApp.

5.1.4 Design Implications

- Snapshots can be managed at both the vApp and individual virtual machine level.
- Reverting an individual virtual machine to a snapshot does not restore vApp level constructs such as OVF properties or network mappings/properties.
- vCloud Director snapshots apply both to vApps that are thin provisioned and fast provisioned.

5.2 Storage DRS with vCloud Director

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

5.2.1 Background

VMware has introduced storage clusters and VM Storage Profiles in previous versions of vSphere, but vCloud Director has never been able to consume them.

Starting with vCloud Director 5.1 you can now expose VM Storage Profiles inside vCloud Director virtual datacenters thus allowing the end user to choose from multiple tiers of storage within the same virtual datacenter.

Figure 34 outlines the high level architecture of how vCloud Director can map these features available in the core platform.

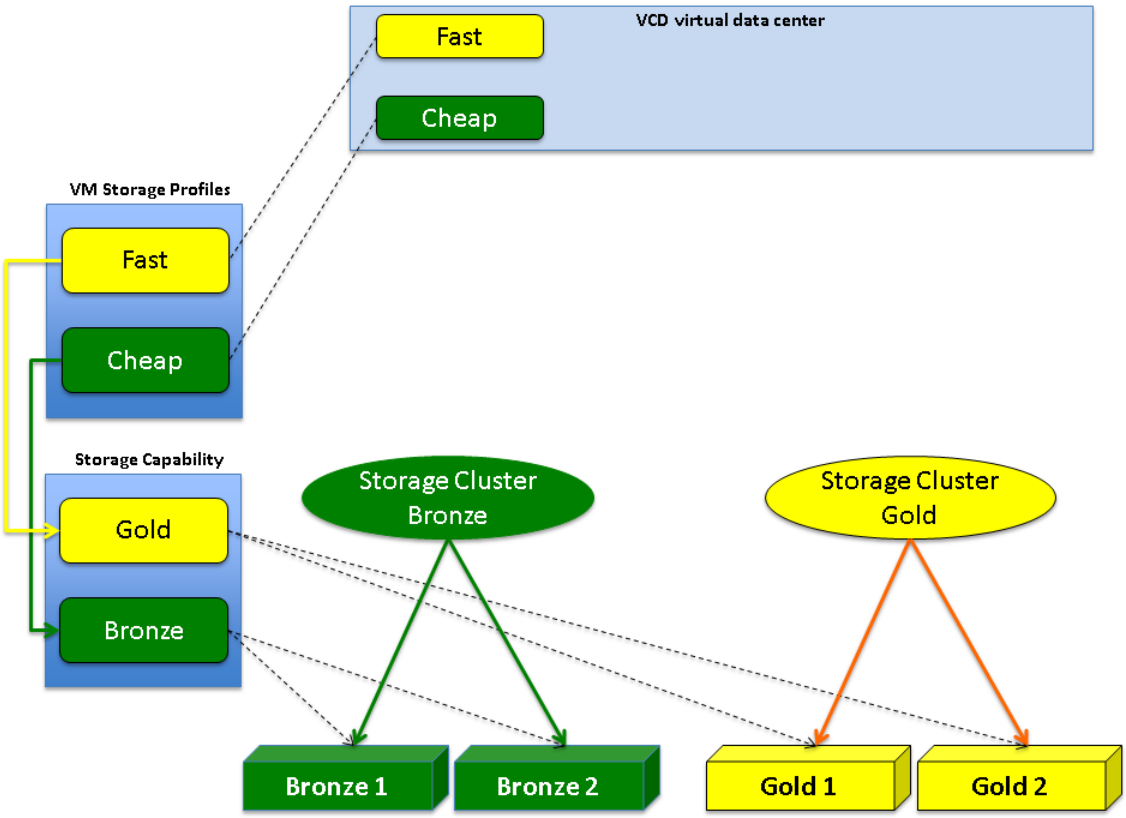
5.2.2 Use Case

The vCloud admin would like to provision a virtual datacenter to an organization and configure two storage profiles to be able to deploy vApp workloads in the proper storage class according to particular requirements. The vCloud admin would also like to create storage profiles that leverage datastore grouping features at the platform level so that the platform can automatically choose the best datastore in a cluster of datastores belonging to the same profile.

5.2.3 Example

At a high level, the following is the procedure for a vCloud administrator to produce and expose to the consumer different profiles of storage. This is illustrated in the following figure.

Figure 34. Overview of Storage Profiles Architecture



To produce and expose storage profiles to the consumer

1. Perform the following steps in the vSphere Web Client, as administrator:
   a. Create datastore clusters and select the datastores.
   b. Assign proper storage capabilities to the datastores.
   c. Create VM Storage Profiles.
   d. Declare which storage capability is associated to the VM Storage Profiles.
2. Perform the following steps in vCloud Director, as administrator:
   a. Assign the desired VM Storage Profile to the provider virtual datacenter.
   b. Create an organization virtual datacenter.
   c. Determine a default Storage Profile for the organization virtual datacenter.

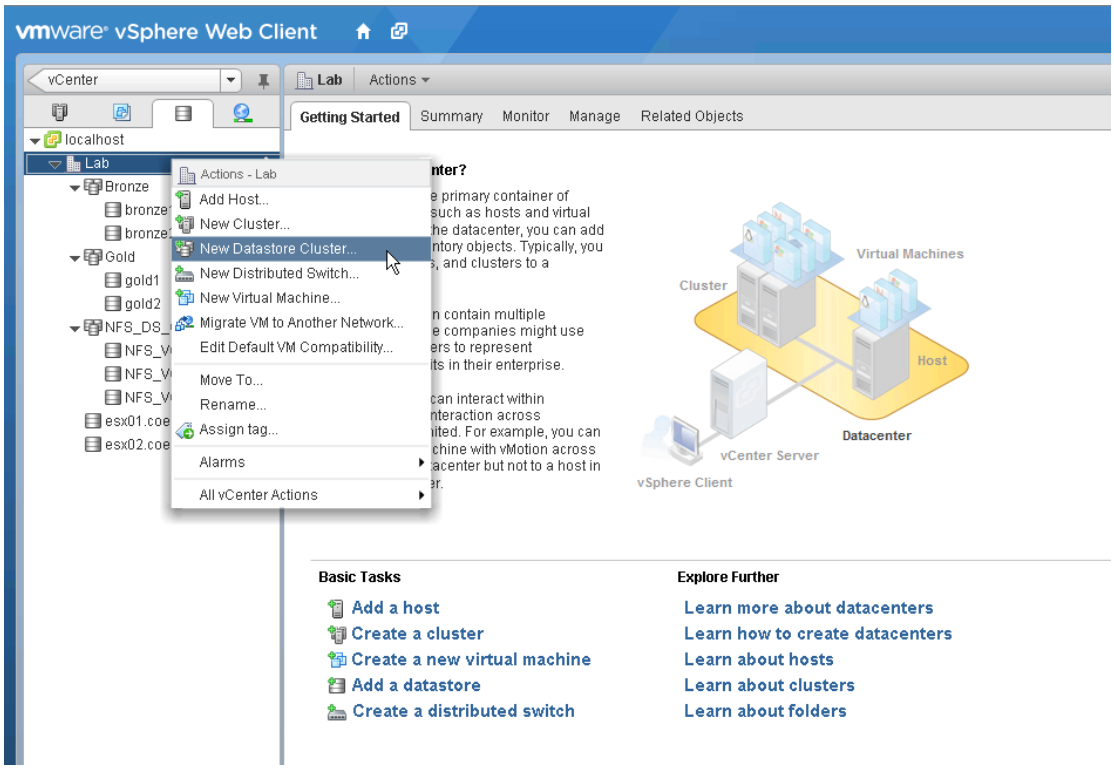
The user is able to consume these classes at vApp deployment time.

5.2.3.1. Implementation

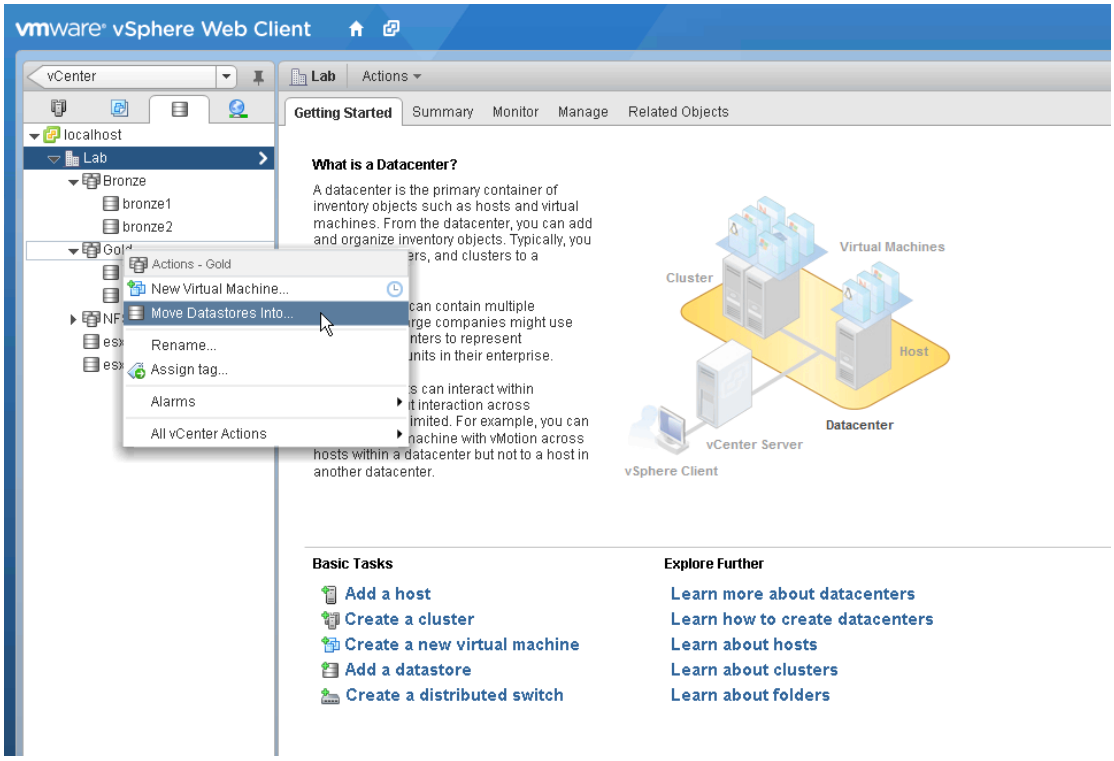
As an example, create Storage DRS clusters in the vSphere domain. In this scenario, there are two datastores (bronze1, bronze2) backed by SATA storage, and two datastores (gold1, gold2) backed by FC storage. There are also other datastores but they are not part of this implementation example.

To create Storage DRS clusters in the vSphere domain

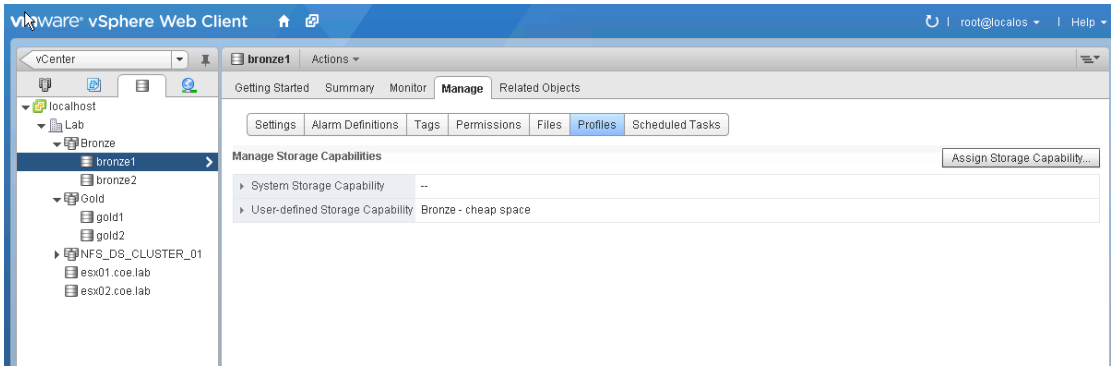
1. Create a new datastore cluster.



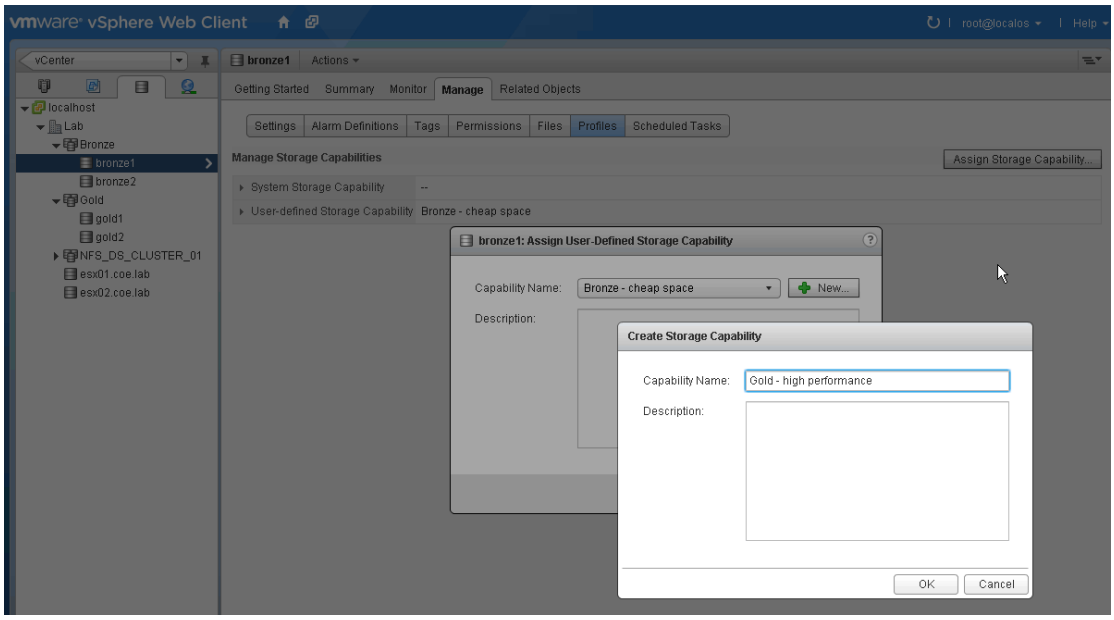
2. Move existing datastores into the cluster.



3. For each of the four datastores, click the datastore and assign a storage capability.

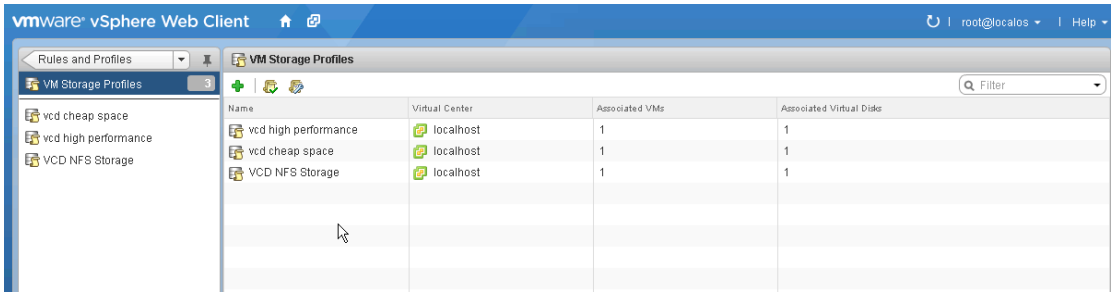


4. If you have not defined a storage capability, create a new one as illustrated in this screenshot. Assign the storage capability. In this example, the storage capabilities are named “Bronze – cheap space” and “Gold – high performance”.



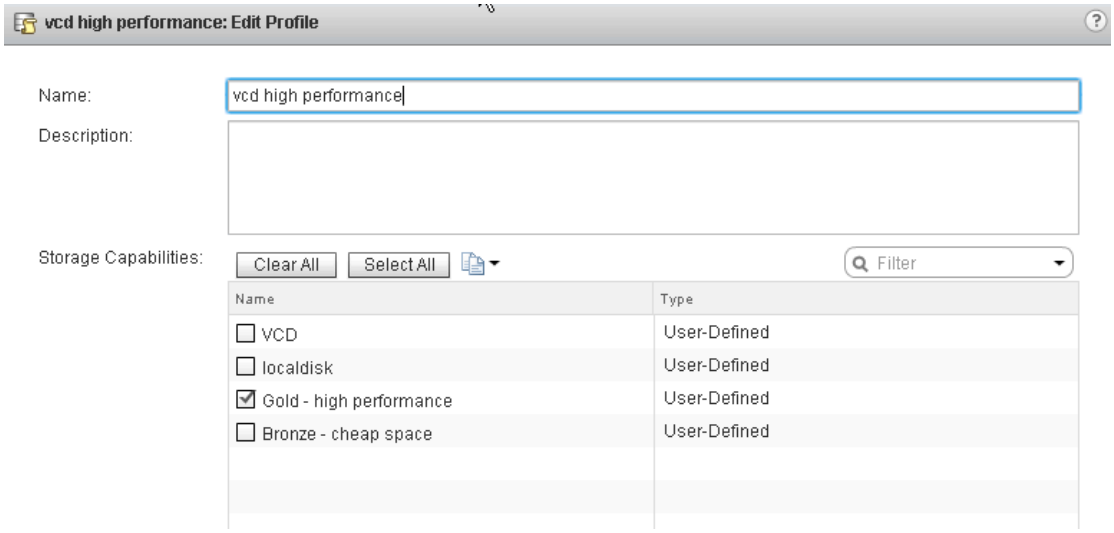
Note: It is important that all datastores in the same cluster are assigned the same homogenous Storage Capability. Otherwise, the proper storage profile does not appear as available in the vCloud Director interface.

5. Create VM Storage Profiles. Each VM Storage Profile can have one or more storage capabilities. This example shows two VM Storage Profiles (“vcd cheap space” and “vcd high performance”).

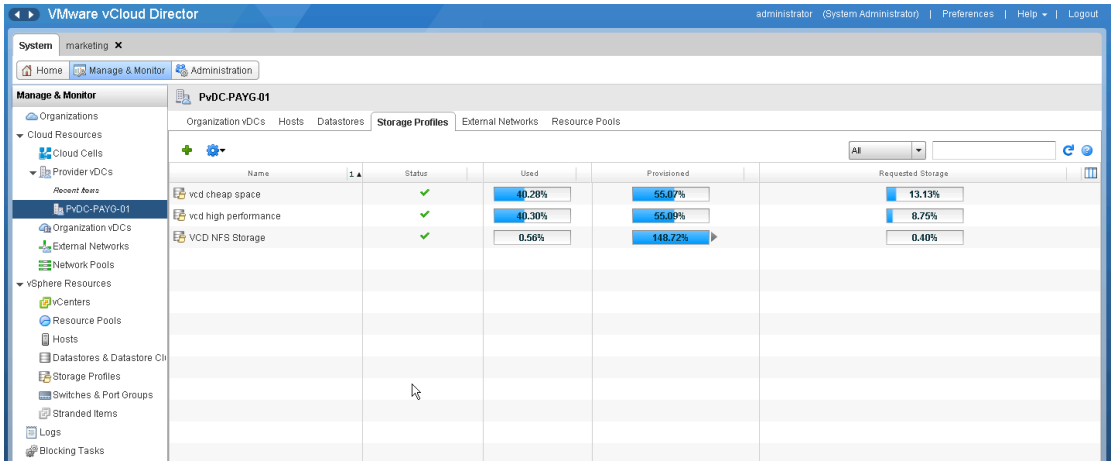


The “Bronze – cheap space” and the “Gold – high performance” storage capabilities are assigned to these VM Storage Profiles, respectively.

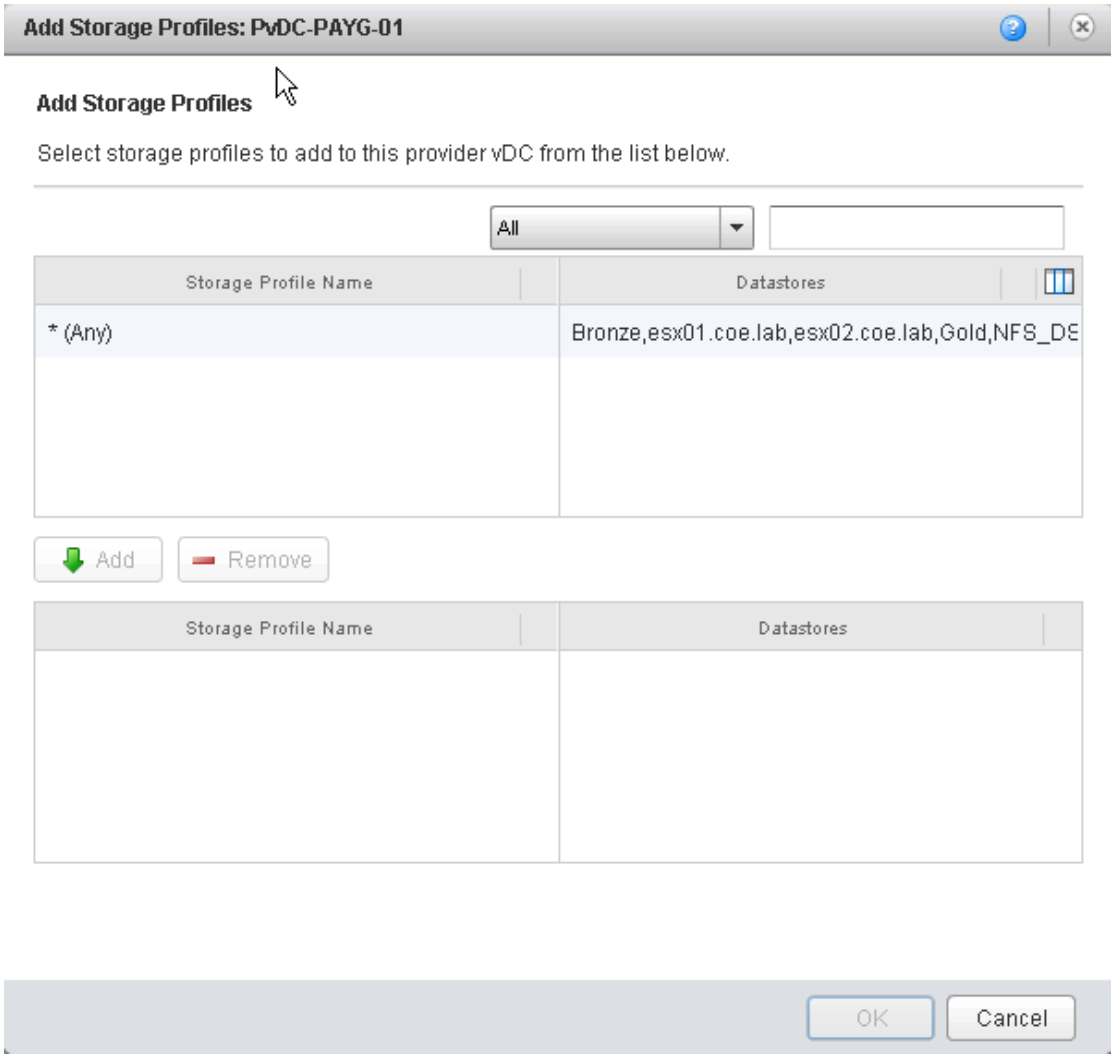
The following screenshot shows how the “vcd – high performance” VM Storage Profile has been configured.



6. Switch to the vCloud Director management portal and configure these VM Storage Profiles in the provider virtual datacenter.

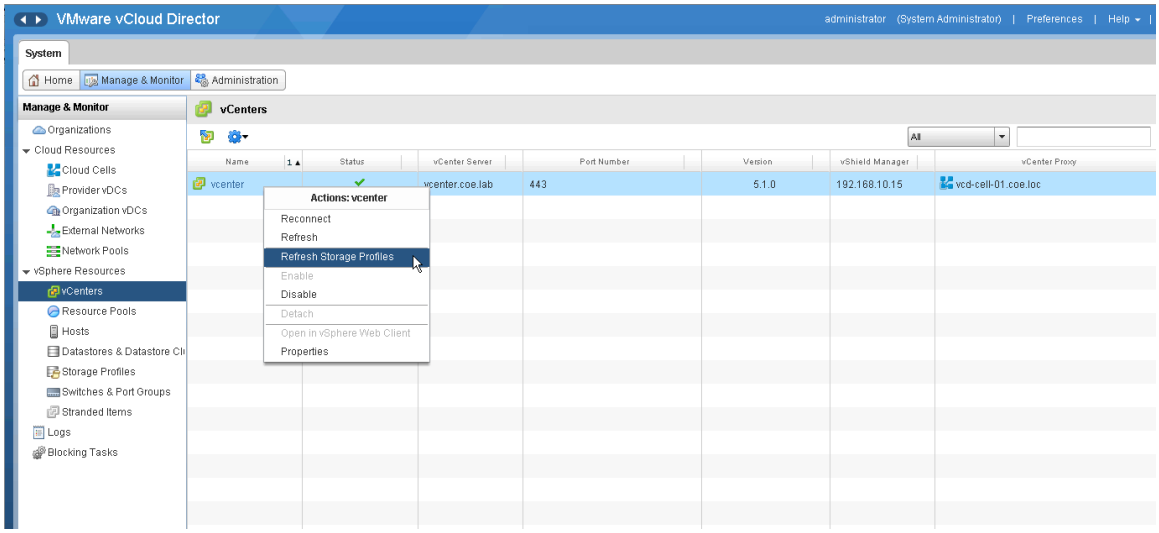


Additional VM Storage Profiles, if available, are displayed in the Add Storage Profile window.

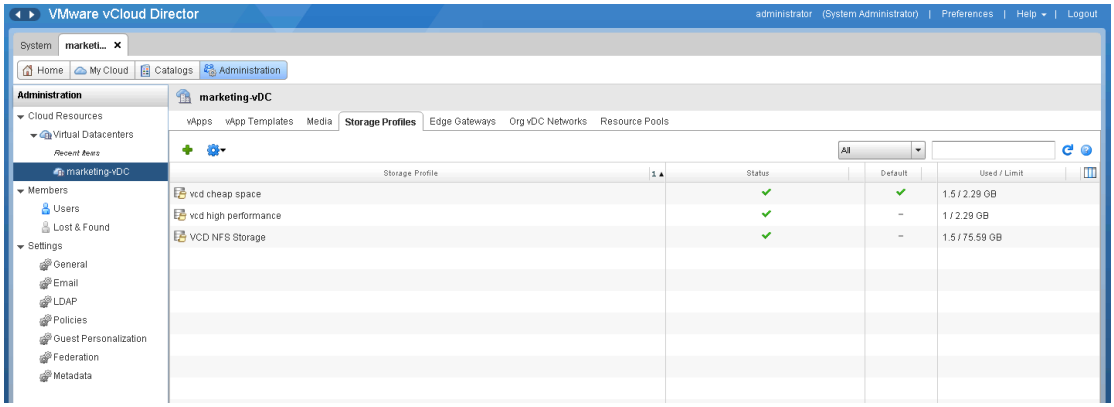


If a configured VM Storage Profile is not displayed in the list, make sure that the proper storage capability has been consistently configured for each datastore in the datastore cluster. Although vSphere marks a cluster as Incompatible when you select a particular VM Storage Profile in the VM deployment wizard, vCloud Director does not show anything in the list unless it is Compatible.

7. A configured VM Storage profile also might not be displayed because vCloud Director has not yet synced with the vSphere platform. This sync happens every five minutes but you can force a re-sync.



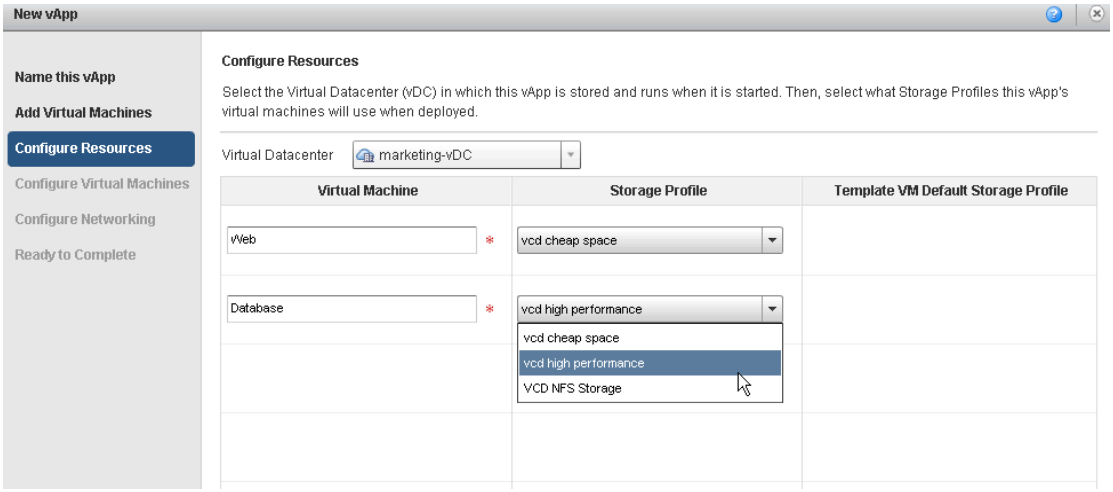
8. Create an organization virtual datacenter for the organization.



Note: This organization virtual datacenter was created for the marketing organization. There is a default Storage Profile defined for the organization virtual datacenter. This is the default selection for virtual machines when vApp Authors and vApp Users deploy virtual machines. Being able to change the default settings of a storage profile is a privilege that you can use when defining user roles. By default, this privilege is assigned to organization administrators.

When a vApp author deploys a two-tier vApp that requires different storage characteristics for each virtual machine in the vApp, the user can choose different storage profiles for each of the virtual machines. This is shown in the following figure.

Figure 35. vApp Deployment Storage Profile



These virtual machines reside on a cluster of datastores managed by vSphere, thus leveraging all of the advantages that the underlying platform provides.

6. Catalog Design Example

6.1 vCloud Public Catalog

Deployment Models: public

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

6.1.1 Background

With vCloud Director, a public service provider can build and offer prepackaged vApps to its tenants through a public catalog. These vApps can vary from basic single virtual machines with only an operating system installed to more complex multitier vApps with application software pre-installed.

6.1.2 Use Case

A public service provider would like to offer to tenants, through a public catalog, a Linux distribution preinstalled in a virtual machine.

This will allow users in any organization to deploy a Linux image in a matter of minutes instead of creating a vApp from scratch and installing the operating system themselves. This is usually a long process with no value-add for the tenants. By providing a basic pre-built operating system the provider's customers can move quickly to adding applications.

Alternatively, the service provider could offer pre-built middleware and application stacks on top of the basic operating system. This is however beyond the scope of this example.

This example shows how to install the Ubuntu server distribution in the virtual machine to be posted in the catalog.

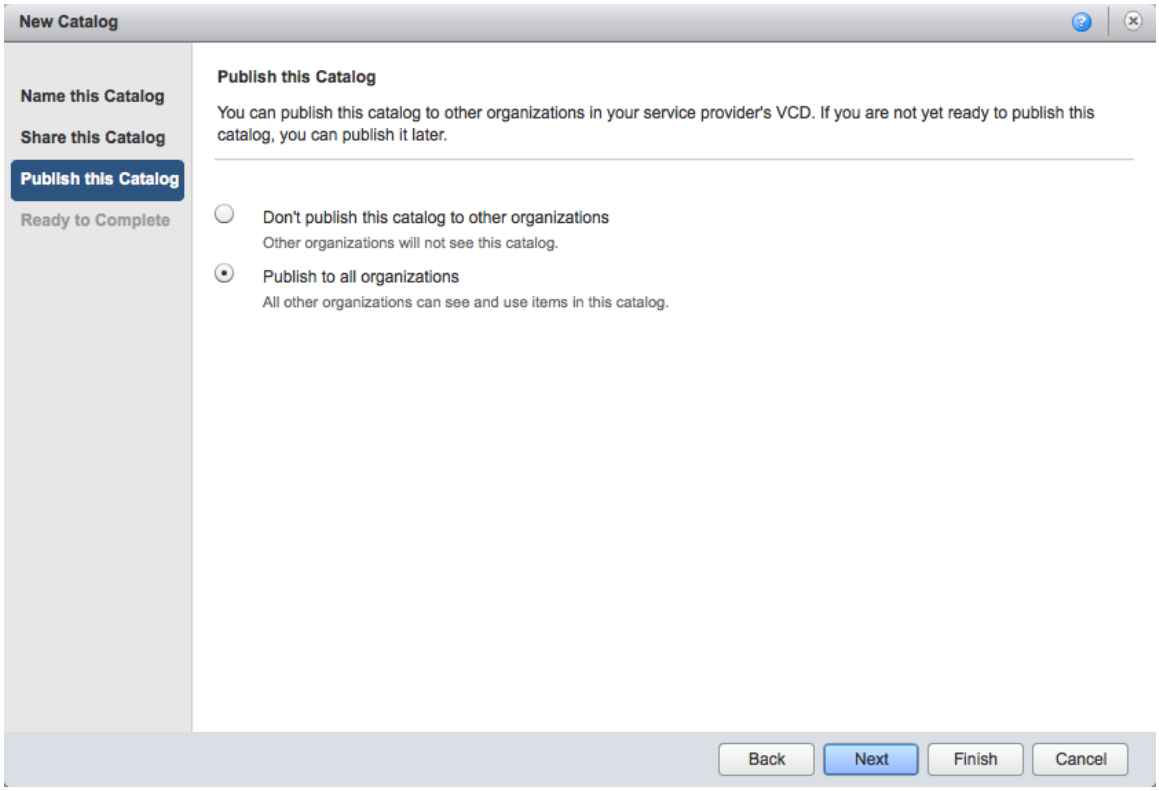
6.1.3 Example

The service provider follows these steps to create a vApp with the Ubuntu Server Linux distribution.

To create an Ubuntu Linux vApp

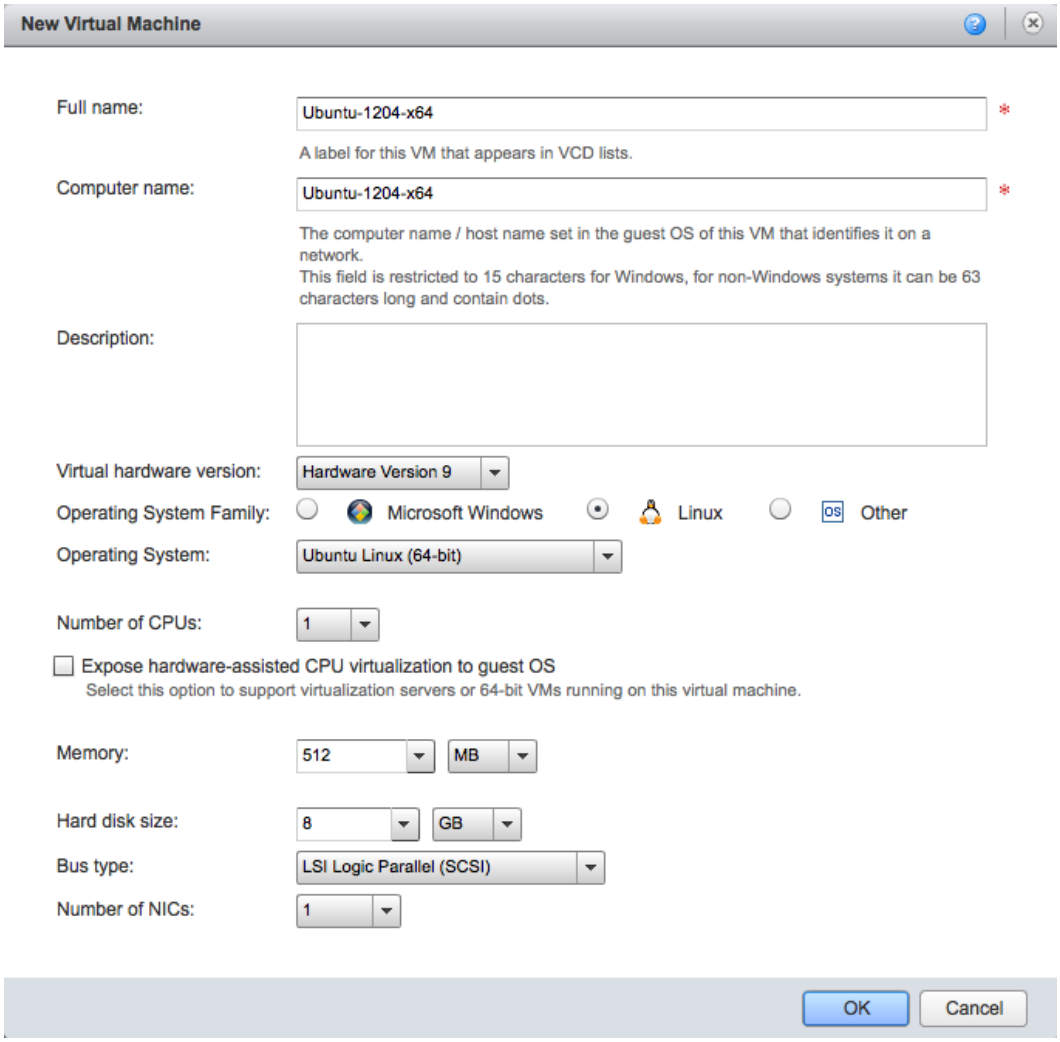
1. Create a VCD service organization whose goal is to export the catalog to all the actual tenants within the cloud offering. A new catalog in this service organization is created and set to "publish to all organizations".

   Note: Although not shown, the "service" organization must be configured to enable Catalog publishing from within the Organization.



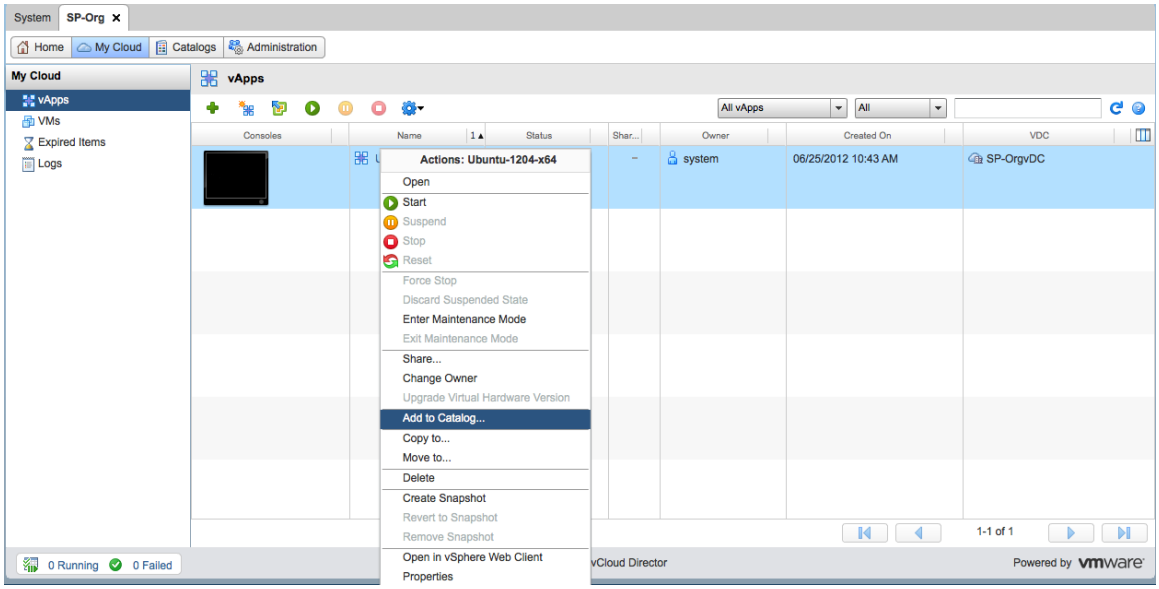
2. Upload the Ubuntu Server 12.04 x64 ISO file. This appears in the media tab of the catalog.

3. Inside this “service” organization, create a new vApp containing a single virtual machine.

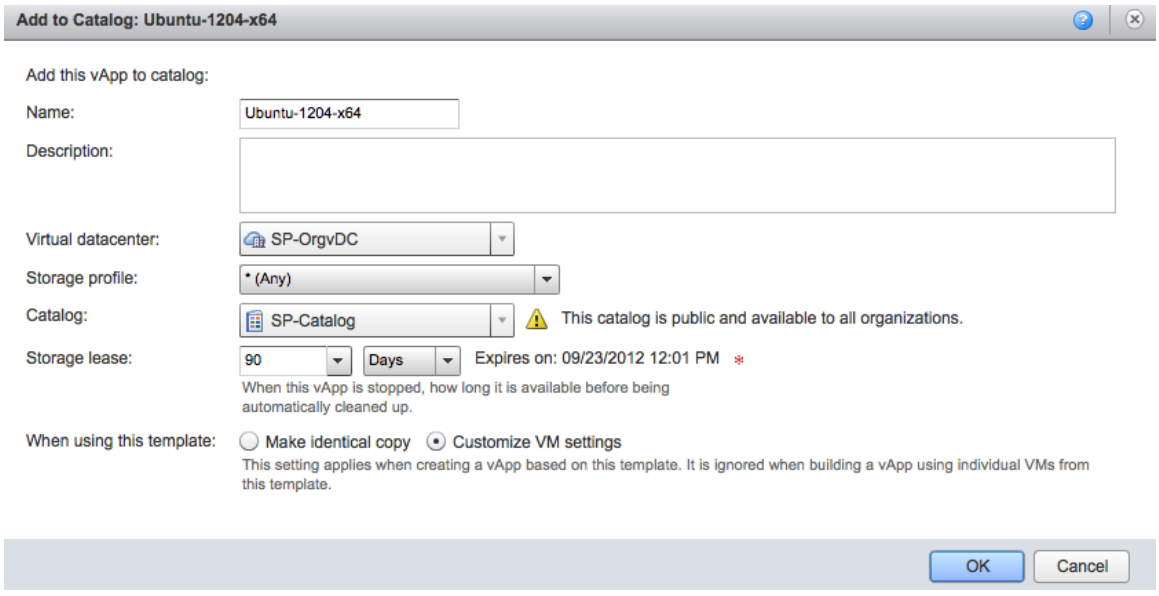


The Ubuntu ISO is mapped to this virtual machine, and the cloud administrator can install the operating system. Eventually the cloud administrator will install VMware Tools inside this virtual machine as described in *Installing VMware Tools in an Ubuntu virtual machine* (http://kb.vmware.com/kb/1022525).

4. After the cloud administrator has completed the basic build operation, the vApp is ready to be put into the catalog of the service organization. Right-click the vApp and select **Add to Catalog**.

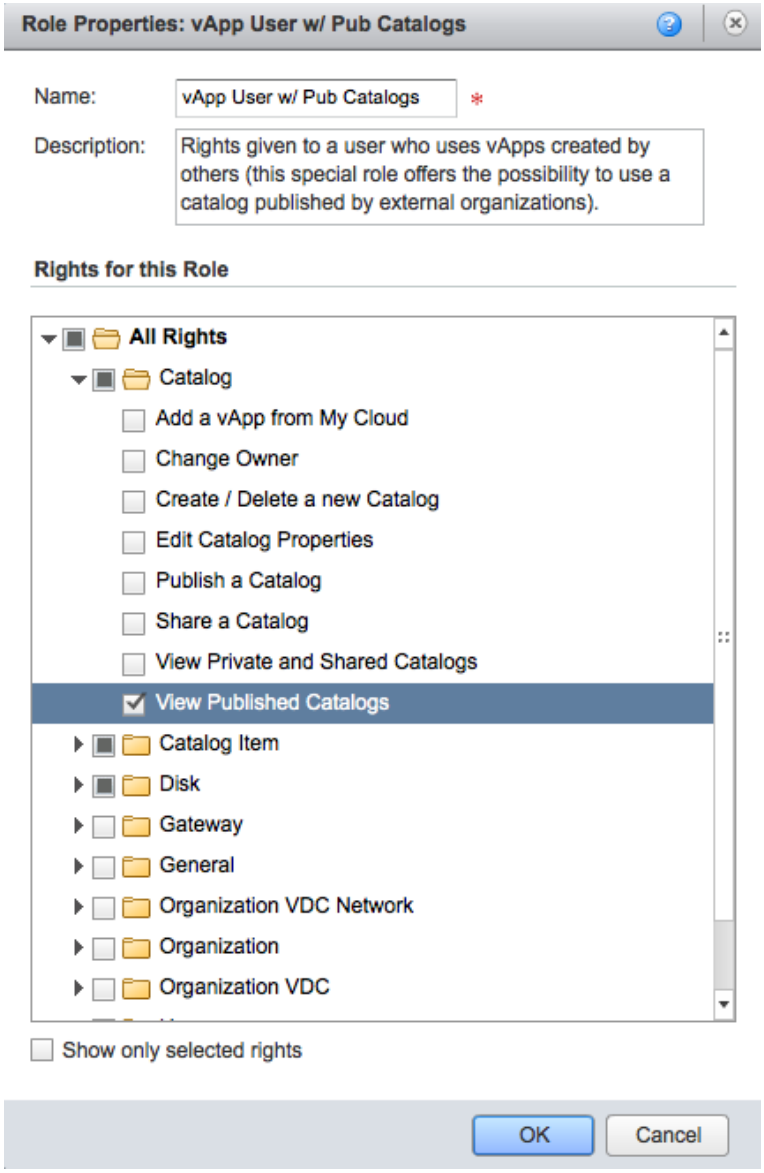


5. Check that this vApp is set to **Customize VM settings**. This allows each user deployment to be unique and the customization of the name of the guest operating system along with its IP address will happen for each deployment from the catalog.



Note: By default, only the organization administrator built-in role has the right to deploy from published catalogs. Other built-in roles such as vApp Author and vApp User do not have this right enabled. This is because it is assumed that the organization administrator goes through the public catalog and copies items from it into the private catalog for its users to consume.

6. In this example, this service provider would like to offer all of the various organization administrators in their clouds the ability to control some of their users' access to leverage the published catalogs. Create a new role on top of the existing default roles. Clone the default "vApp user" role and adds the **View Published Catalogs** right to this newly created role.



This new role now becomes available to the tenants, and specifically to the organization administrators that are creating organization users.

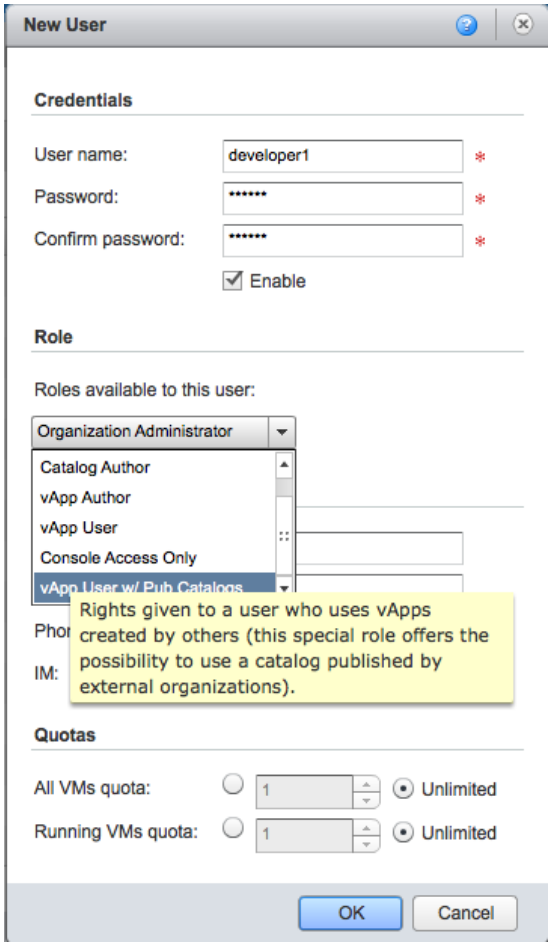
6.1.4 Public Catalog Consumption Example

At this point the service provider can start on-boarding tenants. In this example, an organization, called Org1 represents a new customer that has subscribed to the IaaS cloud this service provider offers.

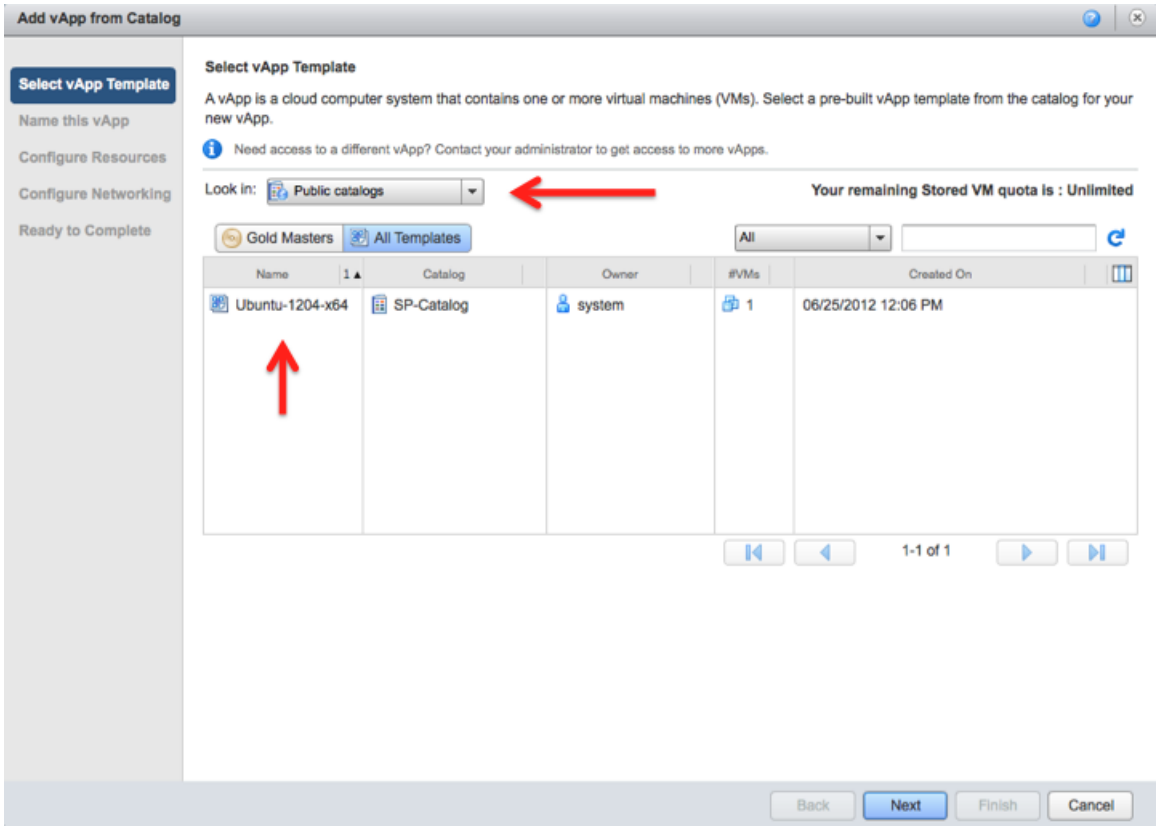
This organization does not yet have a local catalog. If desired, organization administrators can create local catalogs later that can be used as private repositories of their own templates after customization.

To create a local catalog

1. Create a new user (called **developer1**) and assign the newly created custom role **vApp user w/ Public Catalogs** to this new user.



This allows the developer1 user to have access to the templates the cloud provider has shared in the public catalog.



2. The user developer1 can deploy this template to the user's cloud as single virtual machine, or as part of a multi-machine vApp.

7. vCloud Security Examples

7.1 Single Sign-On (SSO) – Provider

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1, SAML Compliant Identity Provider (IdP).

The primary purpose and role of the IdP is to manage the identity information and provide a central authentication service to trusted service providers.

7.1.1 Background

Support for single sign-on (SSO) in the cloud environment has become a necessity, as there are many different management applications that a service provider and enterprise customer typically use. Some of these applications are part of the platform, and others are delivered by third parties but should be integrated in the cloud solution.

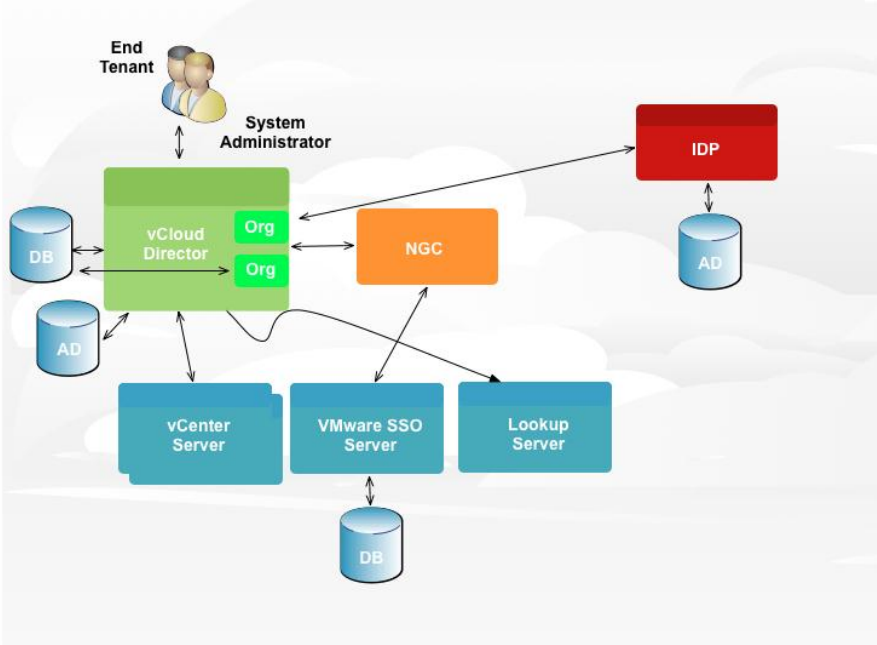
The identity and federation market has moved from a closed enterprise-centric view to an open federated view. Not only do service providers and enterprise customers alike expect single sign-on across applications within the client environments, but they would also like the same identity to work across security boundaries in public cloud setups as well as with SaaS applications. In a private and public cloud setup, the authentication service must support multitenancy as well.

One of the cornerstones of achieving federation is the ability to make user identities transportable from one security domain to another relatively seamlessly. The industry has adopted standards such as WS-Trust and SAML for achieving this. VMware adheres to these standards and builds a Secure Token Service (STS) that generates SAML 2.0 tokens. These standards are also very important for supporting multisite use cases because this allows for Cloud components like vCenter to be passed a SAML token from a previously authenticated secure session. As long as there is mutual trust between the Cloud environments the same authenticated SAML token is respected.

7.1.2 Use Case

In this use case of the Service Provider SSO, a vCloud administrator provides credentials to the UI client only once, which validates them against the SSO server. If the validation is successful, the SSO server issues a SAML token, which then can be used by the UI client to access both vCenter and vCloud Director without having to enter credentials multiple times. The logical architecture for this is shown in Figure 36.

Figure 36. Cloud Provider SSO Logical Architecture

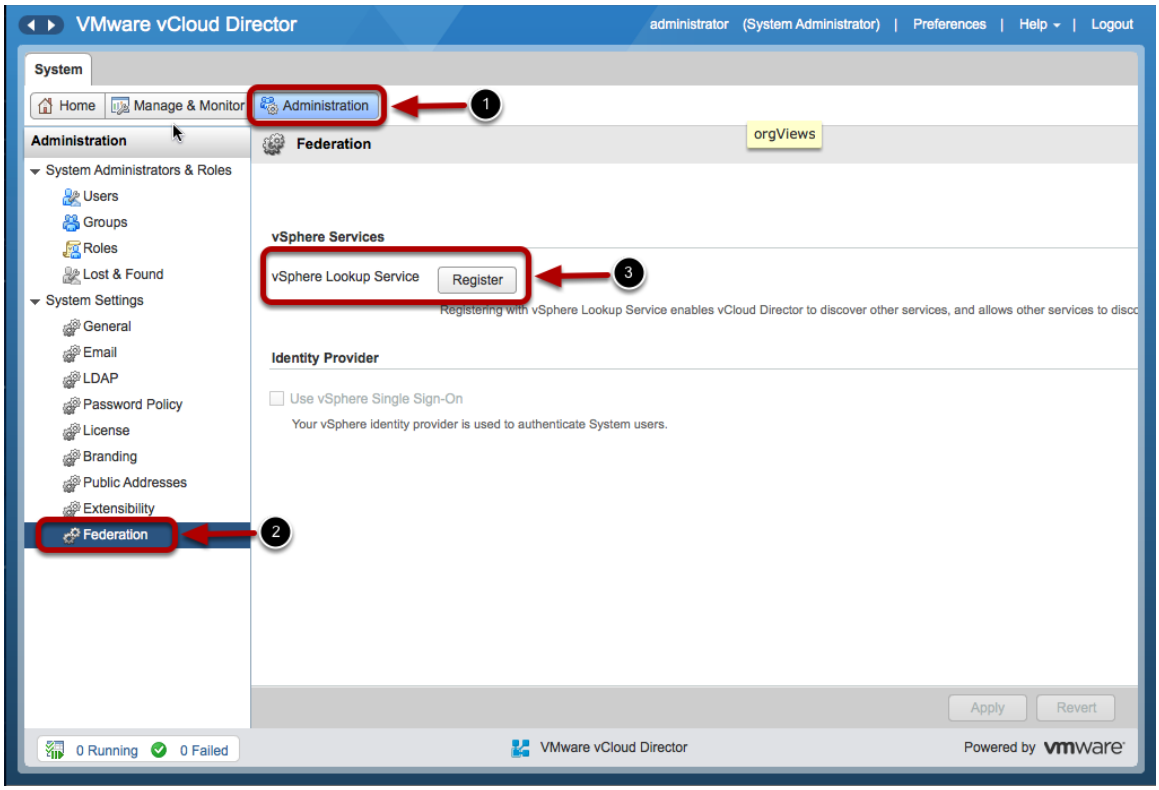


7.1.3 Example

This example shows how vCloud Director administrators who are already authenticated to the vCenter Server through the vSphere Web Client do not have to separately authenticate to vCloud Director.

To authenticate through vCloud Director with single sign-on

1. Log in to vCloud Director as the administrator.
2. Register vCloud Director with the Lookup Service. Go to **Administration** > **System Settings** > **Federation** tab, and click the **Register** button under **vSphere Services**.



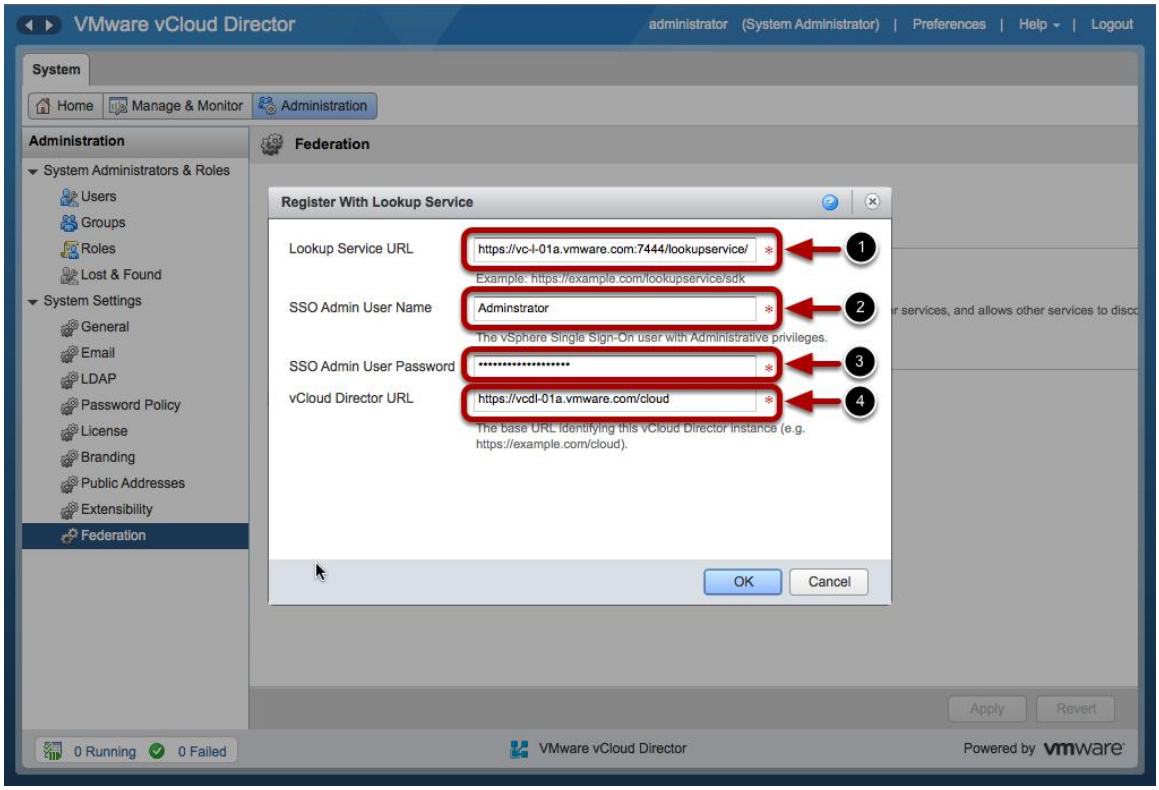
3. Enter the **Lookup Service URL** for the vCenter server you want this vCloud Director to SSO with, as follows:

   <qualified domain name of vcenter-server>:7444/lookupservice/sdk

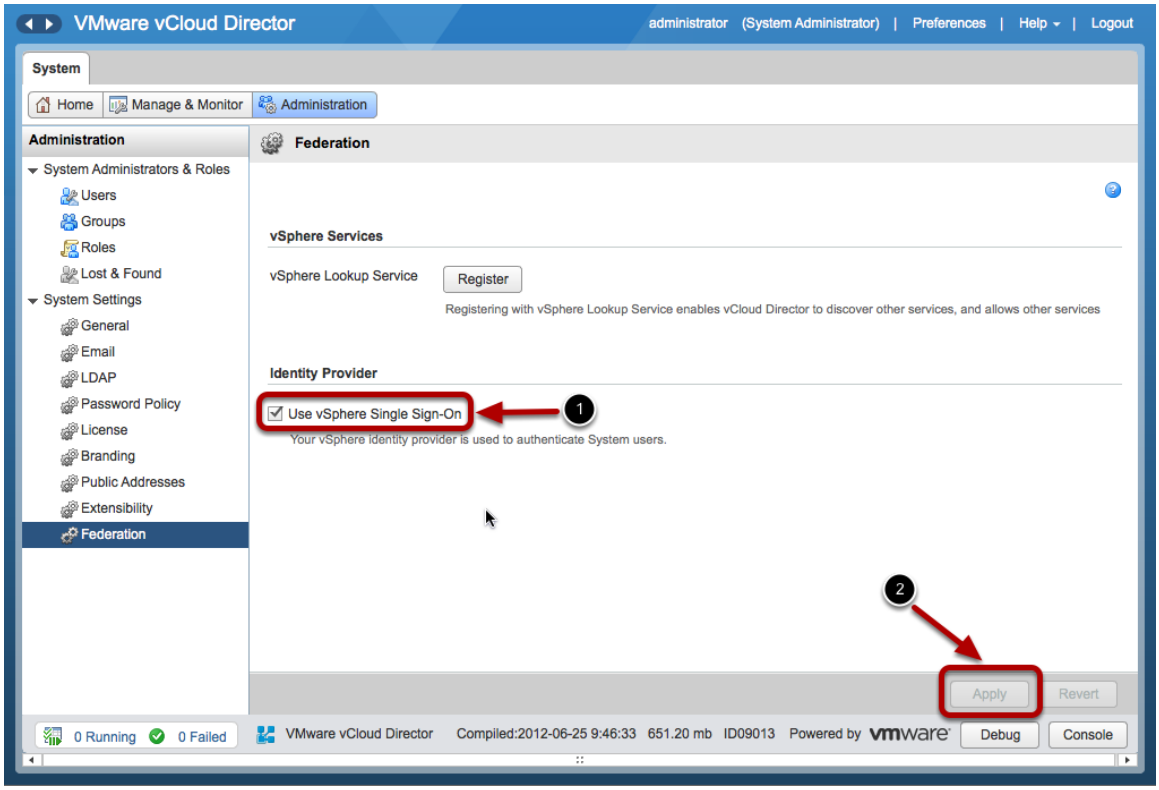
4. Enter the **SSO Admin User Name** and **SSO Admin User Password**.
5. Enter the **vCloud Director URL** as follows:

   <qualified domain name of vcd-server>/cloud

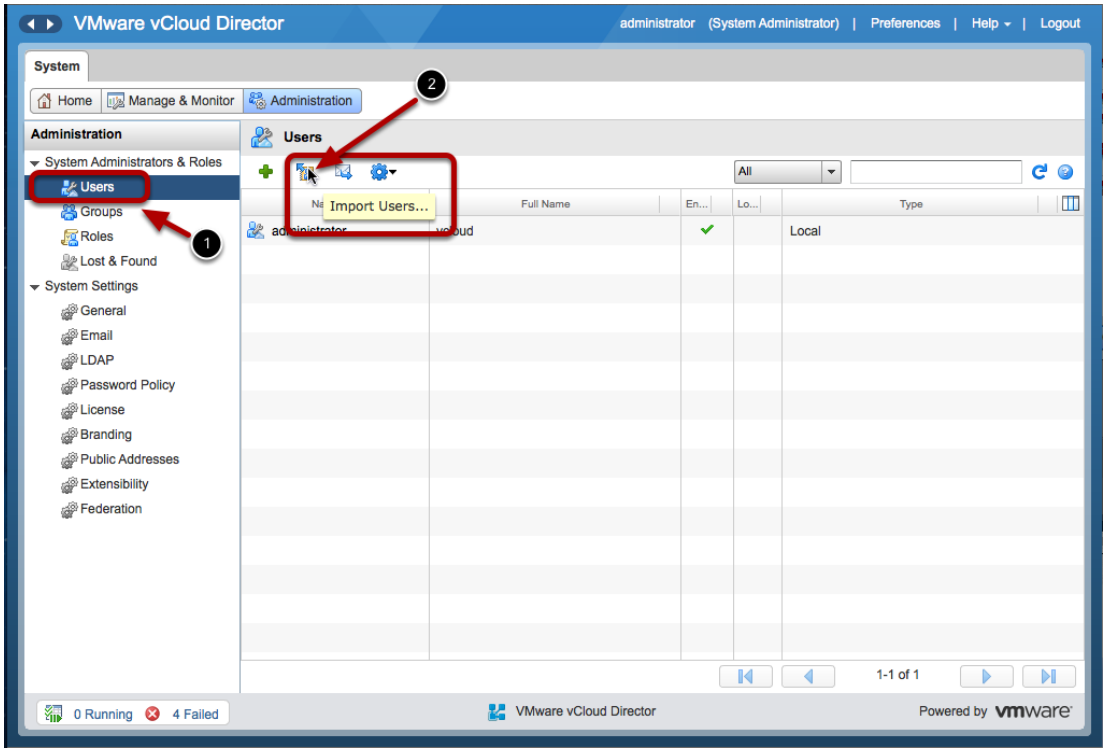
6. Click **OK** and wait for the dialog box to be dismissed.



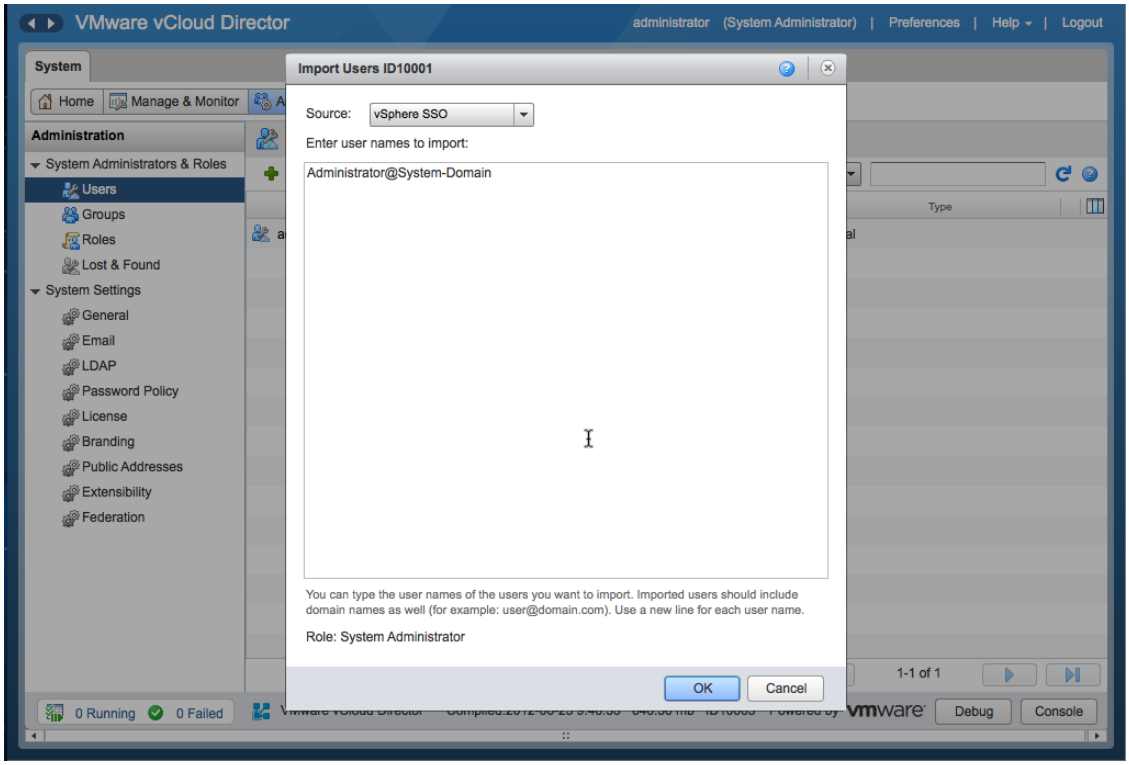
7. Select **Use vSphere Single Sign-On** and click **Apply**.

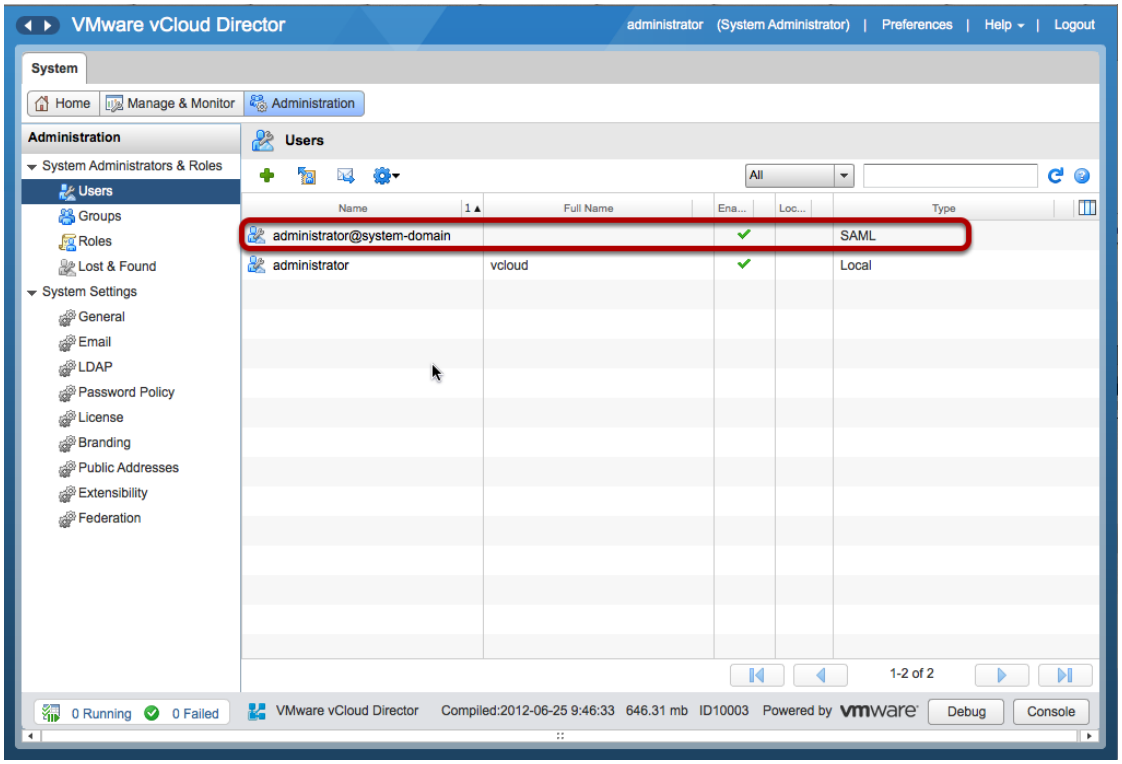


8. Click **Users** and import a vSphere SSO user into vCloud Director.

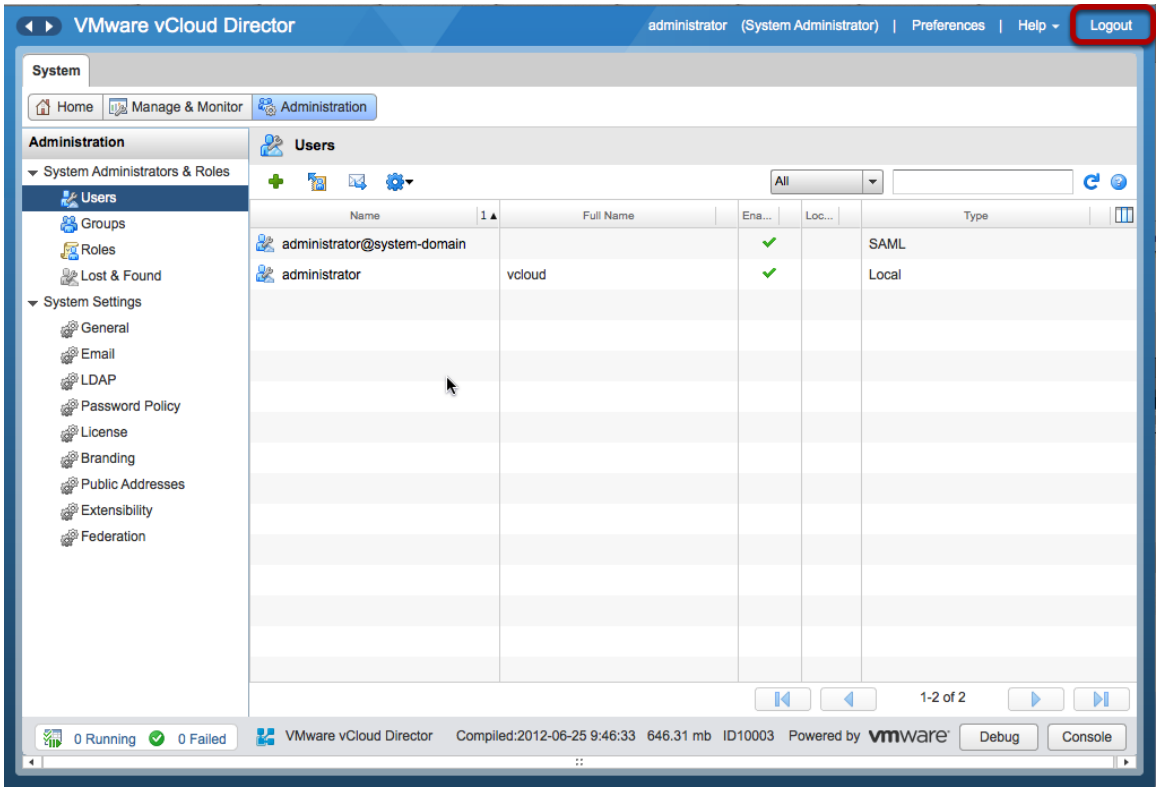


9. As an example, import the **Administrator@System-Domain** user from vCenter SSO server.

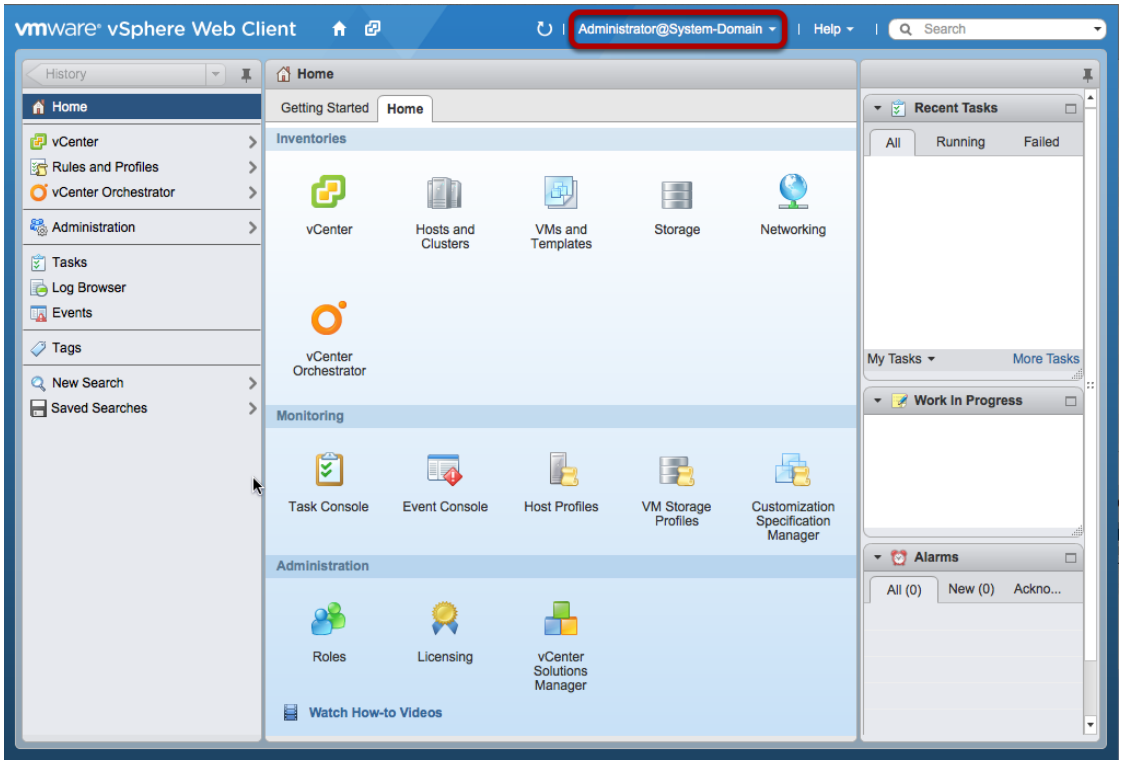




10. Log out from vCloud Director.



11. Go to the vCenter Sever and log in as the user imported in a preceding step.



12. Open a new browser tab and go to vCloud Director. You are logged in without requiring further authentication.
13. To log in as a vCloud Director local user, go to:

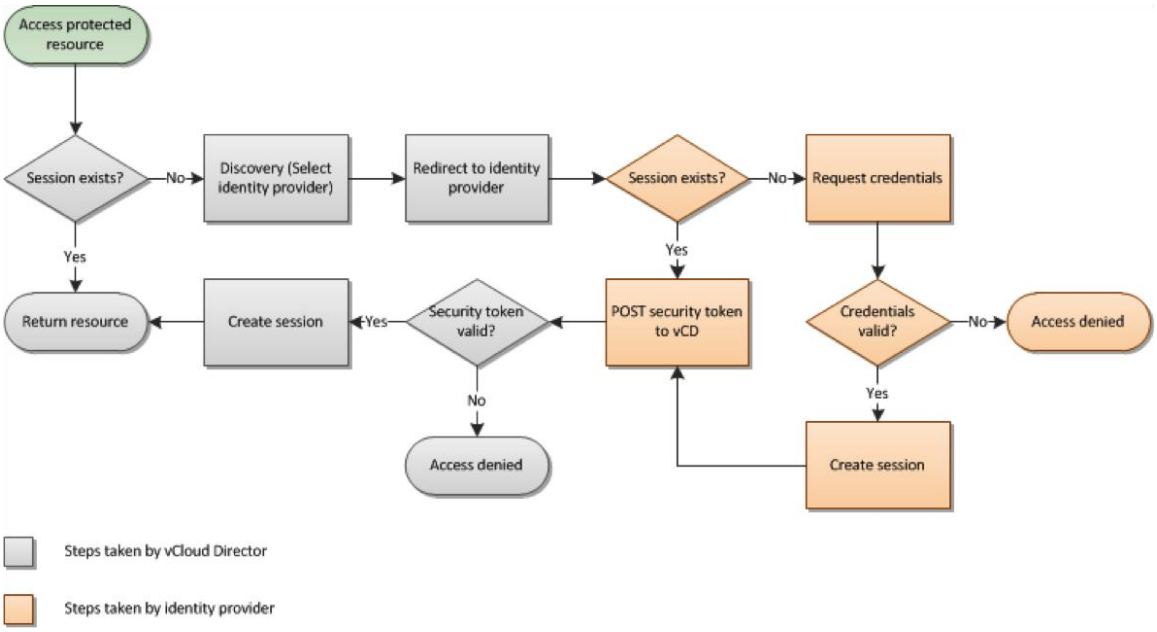
    https://<vcd-server>/cloud/login.jsp

14. Using RDP, go to the virtual machine where the vSphere Web Client is running.
    a. Open a new browser tab and go to vCloud Director.
    b. You are redirected to the vSphere Web Client where you can log in as **Administrator@System-Domain**.
    c. Upon successful authentication you are redirected to vCloud Director.

7.1.4 Single Sign-On Authentication Workflow

The following figure illustrates the workflow for a Single Sign-On session using vCloud Director and the identity provider.

Figure 37. Single Sign-On Authentication Workflow



7.1.5 SSO and Authenticating with the vCloud API

You can use the `POST/sessions` vCloud API, which accepts security tokens as the request body:

- HTTP-Basic authentication: Logs in using the user name and password to the integrated identity provider for backwards compatibility with vCloud Director 1.5.
- SAML assertion: Verifies that the assertion is trusted.
- Proprietary token: Verifies that the token from the integrated identity provider is valid.

You can use the vCloud API `GET /org/{id}/hostedIdentityProvider/token` which returns the security token for the integrated identity provider:

- HTTP-Basic authentication logs in using the user name and password.
- Kerberos: Verifies a Kerberos token using Active Directory settings.

You can use the vCloud API `GET /org/{id}/identityProviders` which returns a list of IdPs federated with vCloud (the currently integrated identity provider and possibly an external identity provider), which can be called anonymously.

You can use the vCloud API `GET /org/{id}/saml/authnRequest` which returns the signed SAML AuthnRequest.

7.1.6 Design Implications

- Use single sign-on (SSO) to provide a common service, both internally and externally.
- Single sign-on (SSO) can be combined with the use of smart cards or Common Access Cards (CACs) for initial authentication to a directory service.
- You must use a supported Identity Provider (IdP):
  - Identity sources: OpenAM, Active Directory Federation Services, Shibboleth
- Deployment Models: Single mode (one node), HA mode (multiple nodes), Replication mode.
- Use a high availability architecture to provide a highly available single sign-on (SSO) service.

Deploying vCenter Single Sign-On as a cluster means that two or more instances of vCenter Single Sign-On are installed in high availability (HA) mode. vCenter Single Sign-On HA mode is not the same as vSphere HA. All instances of vCenter Single Sign-On use the same database and should point to the same identity sources. Single Sign-On administrator users, when connected to vCenter Server through the vSphere Web Client, see the primary Single Sign-On instance. In this deployment scenario, the installation process grants admin@System-Domain vCenter Server privileges by default. In addition, the installation process creates the user admin@System-Domain to manage vCenter Single Sign-On.

7.2 Single Sign-On (SSO) – Consumer

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1.

7.2.1 Background

The security and identity infrastructure in the cloud has become an important management platform component. Without support for single sign-on (SSO) in the cloud infrastructure or the ability to support federation, every cloud solution would need to create its own user identities to participate in the management process, which would dramatically increase the administrative overhead.

7.2.2 Use Case

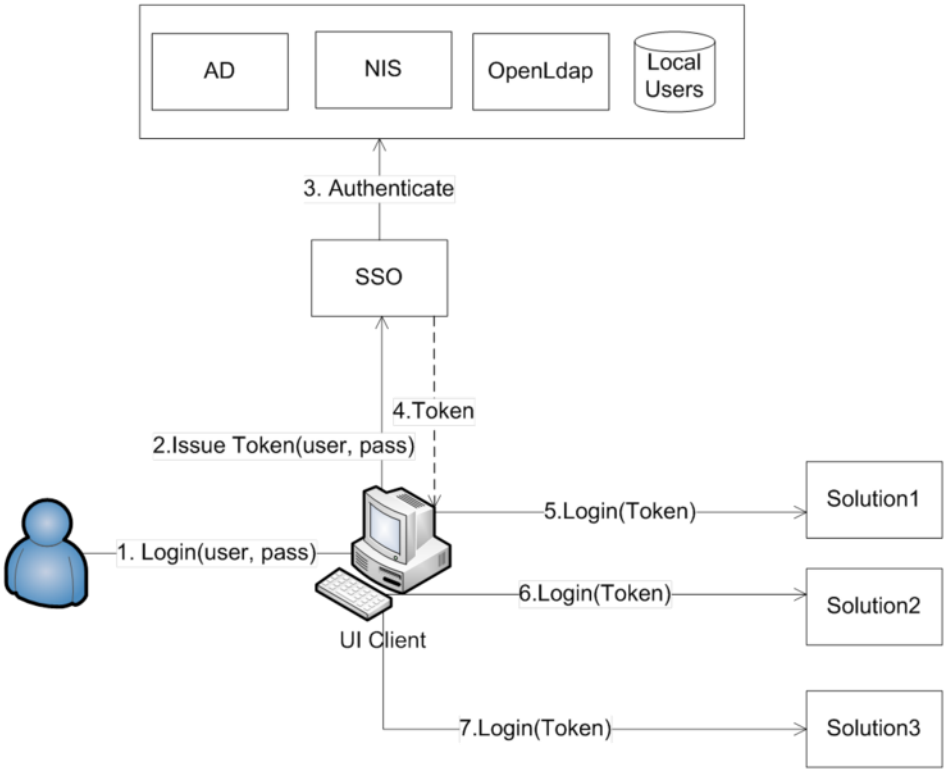
The Web single sign-on (SSO) feature and configuration are exposed through vCloud Director and can be used in both service provider and consumer architecture. There are several use cases that can be used, as follows:

- Between a single client and multiple back end servers.
- Solution-to-solution authentication.
- Delegation.
- Delegation and renew.

7.2.2.1. Between a Single Client and Multiple Back End Servers

The classic single sign-on (SSO) use case is the single sign-on between a single client and multiple back end services. A user accesses multiple back end servers through a single UI client. The user provides credentials to the UI client only once, which validates them against the SSO server. If the validation is successful, the SSO server issues a SAML token which then can be used by the UI client to access other back end servers. The following figure illustrates this use case.

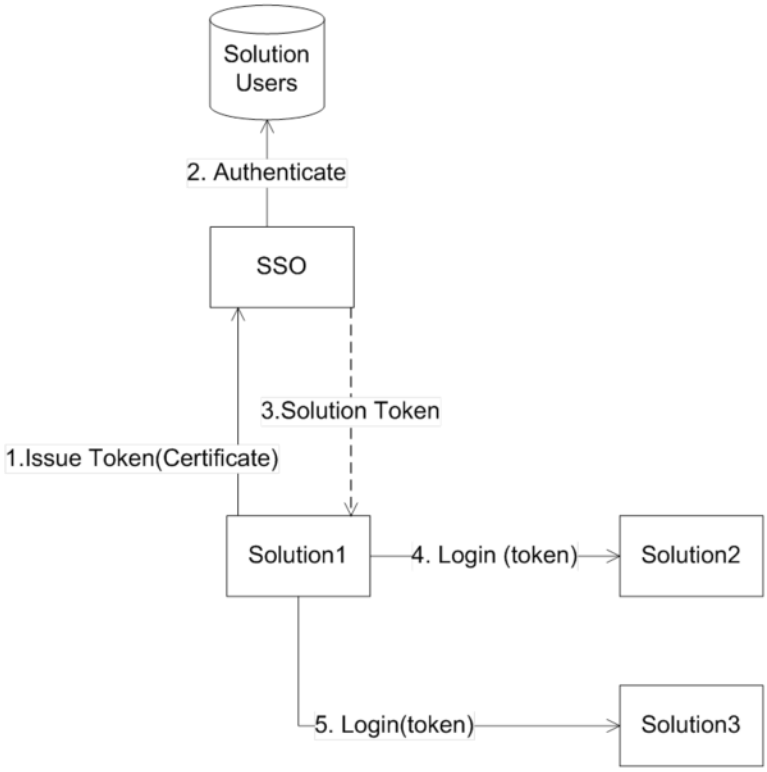
Figure 38. Single Sign-On (SSO) Between a Client and Multiple Back End Services



7.2.2.2. Solution-to-Solution Authentication

With solution-to-solution authentication, the goal is to assign an SSO user to each of the solutions. In this use case two solutions communicate with each other. Before they start to communicate they must prove each other's identity. The solution that initiates communication requests from the SSO server issues a SAML token which asserts its identity. As part of this request the solution proves its identity using its own private key. After the SSO server has issued a token the solution can use that token to access any other solution as if it is a normal user. For this use case to work, each solution must be registered with its public key in the SSO server. The following figure illustrates this use case.

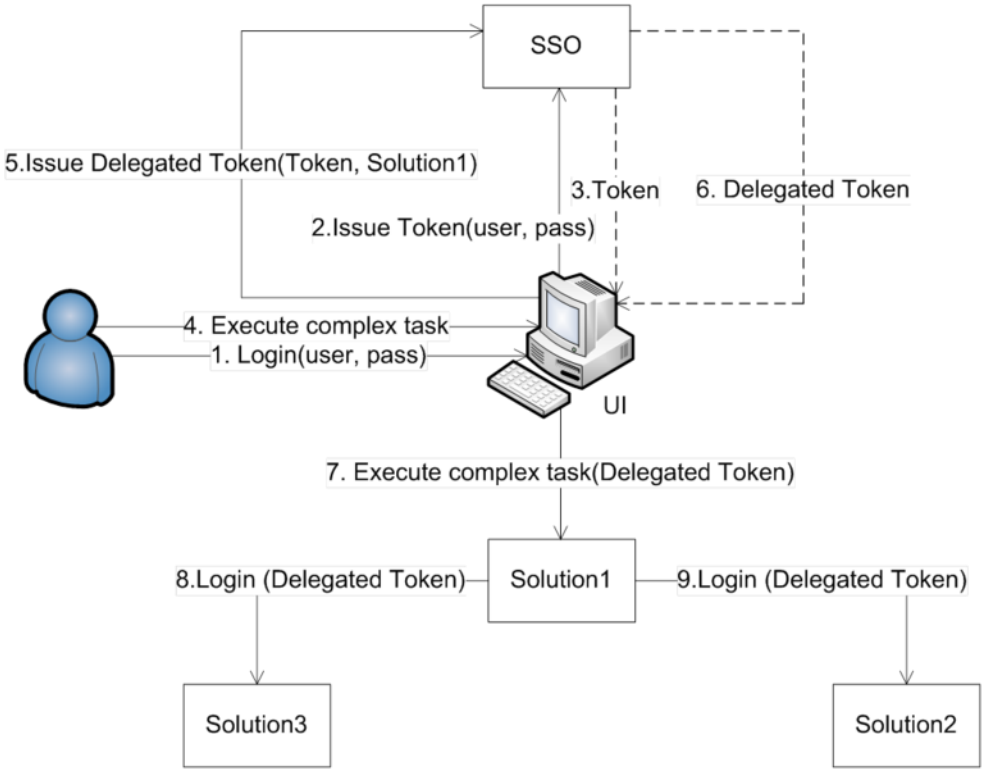
Figure 39. SSO Solution-to-Solution Authentication



7.2.2.3. Delegation

Executing tasks on behalf of a user is referred to as delegation. In this example use case, some workflows, which an end user initiates, might require multiple solutions to communicate with each other. This use case shows the SSO support for such work flows. Before the user can initiate the workflow through a given UI, the user must provide credentials. The UI then validates those credentials against the SSO server, which issues a SAML token. Then the user decides to initiate a workflow, which requires Solution-1 to access Solution-2 and Solution-3 on behalf of the end user. As part of this process, the UI requests from the SSO server a so-called “delegated” token for Solution-1 by providing the SAML token of the end user. The delegated token asserts that the user has granted Solution-1 the privileges to execute tasks on the user’s behalf. After the UI has the delegated token it gives it to Solution-1, which then can use it to log in to Solution-2 and Solution-3. The following figure illustrates this use case.

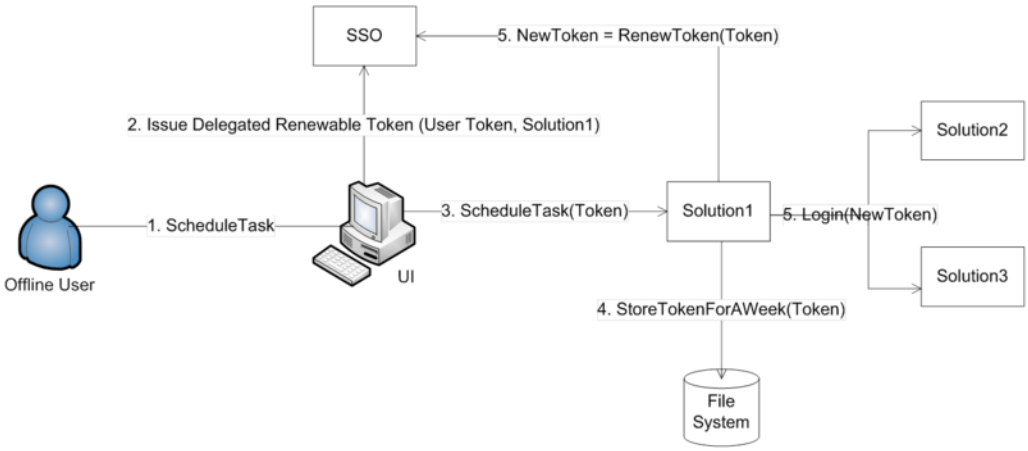
Figure 40. Task Execution on Behalf of a User



7.2.2.4. Delegation and Renew

Delegation and renew defines scheduling a long-lived task. Some long running operations in the infrastructure require long running tasks to be executed in the absence of the end user who has initiated them. The SSO server supports such tasks using delegated and renewable tokens. After a long running task has been identified, the UI obtains from the SSO server a delegated and renewable token. It passes that token to the solution, which performs that long running task. The solution persists the token in a non-secured way, as the token is self-secured. Each time the task gets activated, the solution reads the token from the disk and goes to the SSO server to renew it. By going to the SSO server for the renewal, the solution has the guarantee that the user has not previously deleted it from the system. This use case is illustrated in the following figure.

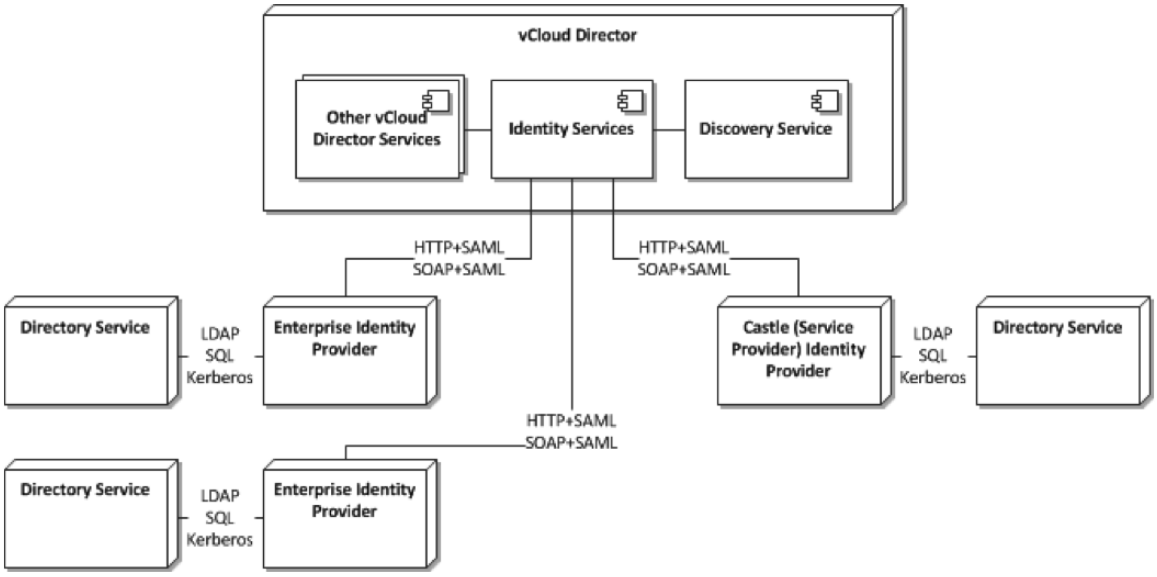
Figure 41. Single Sign-On for Long-Lived Tasks



7.2.3 Example

This section covers a consumer single sign-on deployment architecture, as illustrated in the following figure.

Figure 42. Consumer Logical Single Sign-On Deployment Architecture



This example shows how enterprise customers can log in to vCloud Director with their existing identity management software, whether they are connecting to an internal cloud or to a vCloud Director powered service provider. To demonstrate this behavior you must set up a separate Identity Provider (IdP) using either OpenAM or ADFS. This example uses vCloud Director to create an organization named vCAT and sets the OpenAM IdP as the IdP of that organization. Thereafter, when you log in to your organization, you will be redirected to OpenAM where you can authenticate and be directed back to the vCloud Director portal. This is documented in the following procedure.

To set up single sign-on with OpenAM IdP

1. Set up OpenAM as an enterprise IdP. As an example, using a web browser, go to: http://openam.corp.local:8080/openam/saml2/jsp/exportmetadata.jsp?realm=labs

   This page provides XML text that must be copied and pasted to a text area in vCloud Director in the next step. Right-click the browser window and click **View Source**. Select and copy the entire text and paste it into a text editor such as Notepad. Make sure that there are no blank lines at the top or bottom of the text. Keep this information easily available, as you will need it in the next step (where it is pasted in the organization's Federation settings under the Administration section).

2. Create a vCloud Director organization named **vCAT** and set up the IdP configuration to point to the OpenAM IDP server.
3. Log in to vCloud Director as **administrator**.
4. Create a vCloud Director organization named **vCAT**.
5. Click **Finish** after you enter the name.
6. Go to your organization.
7. Go to **Administration** > **Federation**.
8. Select **Use SAML Identity Provider**.
9. Paste the XML text that was copied from OpenAM in step 1, and **Apply** the changes.
10. Remove any extra spaces in the beginning and end of the SAML text. One way to accomplish this is to remove the XML header at the top up to the opening angle bracket.
11. Go to **users** > **Import** and import some of the users that have been created in OpenAM.
    a. Specify <username>@<domain name>.com in the text area, where <username> is either **orguser** or **orgadmin**.
    b. Assign an organization administrator role to **orgadmin** and **vAppuser** role to **orguser**.
    c. Log out from vCloud Director.
12. Open another browser tab and go to: https://<vcd-server>/cloud/org/Lab/saml/metadata/alias/vcd. This downloads a file called `vcd`. Perform the following steps:
    a. Access openam.corp.local:8080/openam from your browser.
    b. Log in as **amadmin**, password: **<password>**
    c. Go to the **Federation** tab.
    d. Under the **Entity Providers** list, click **Import Entity**.
    e. Select **labs** as the realm name.
    f. Upload the `vcd` file. (Select the first upload button.)
    g. Under the **Circle of Trust** list, click the name of the realm with which you are federating (**labs**).
    h. Under the list of **Available** entity providers, locate the VCD entity. Click **Add**, and click **Save**.
    i. Log out from OpenAM.
    j. Log out from vCloud Director.

13. Type the vCloud Director organization URL: https://<vcd-server>/cloud/org/Lab

    You are redirected to the OpenAM IdP where you can log in as one of the following users:

    - **orgadmin** (password: **<password>**).
    - **orguser** (password: **<password>**).

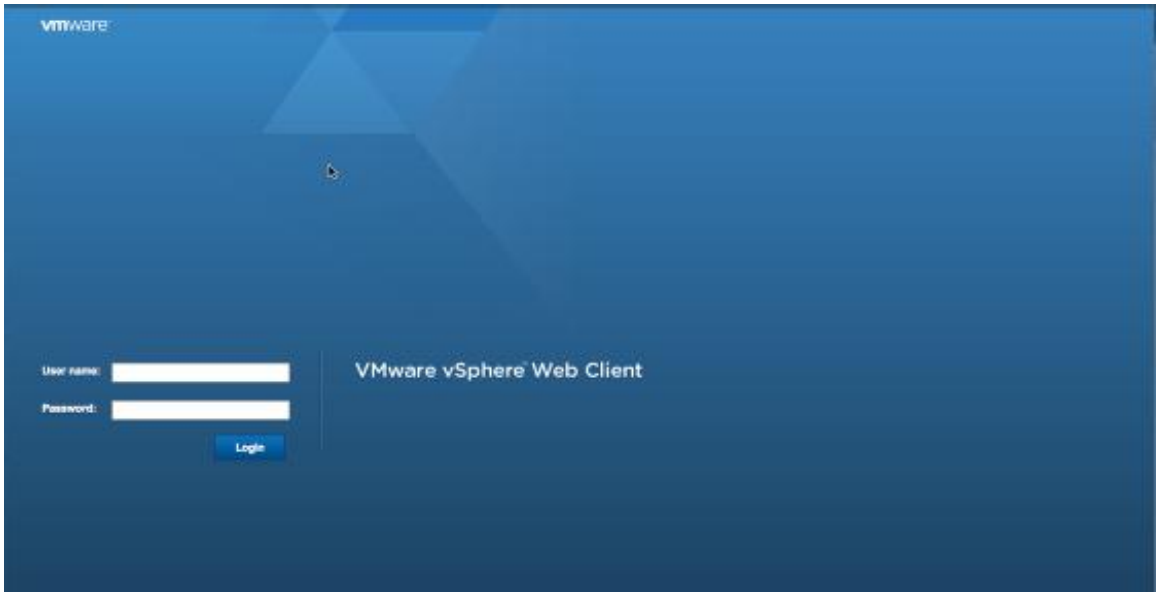
    The user is redirected to vCloud Director after a successful authentication.

7.2.4 Consumer Workflow Example

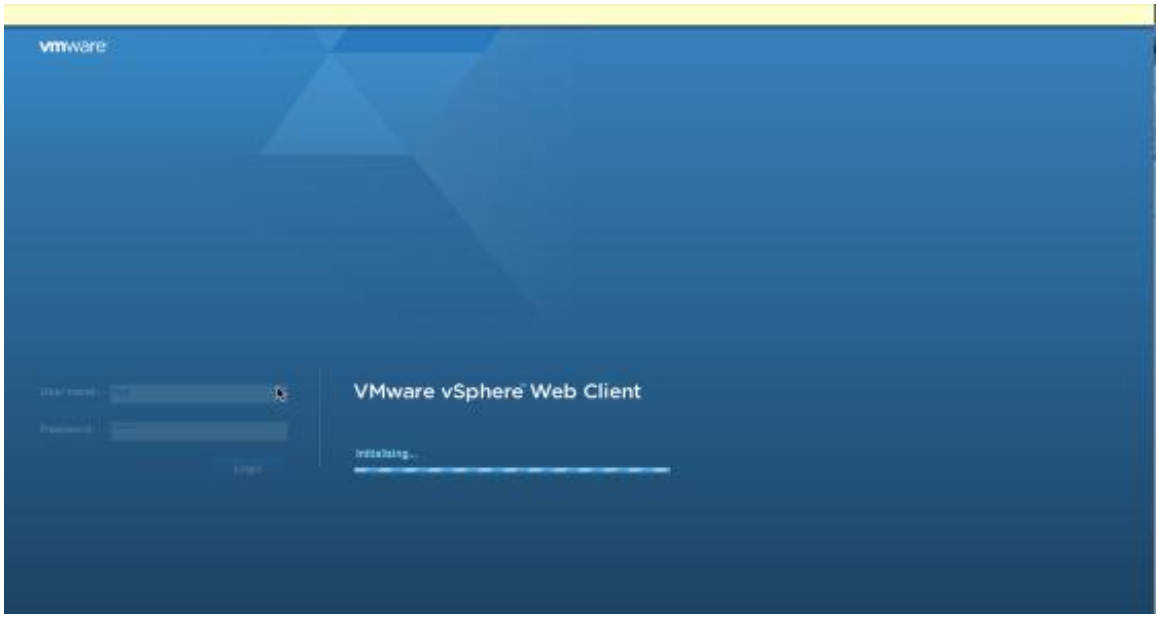
The following gives an example of what happens when you log in in as an end tenant.

To log in as an end tennant

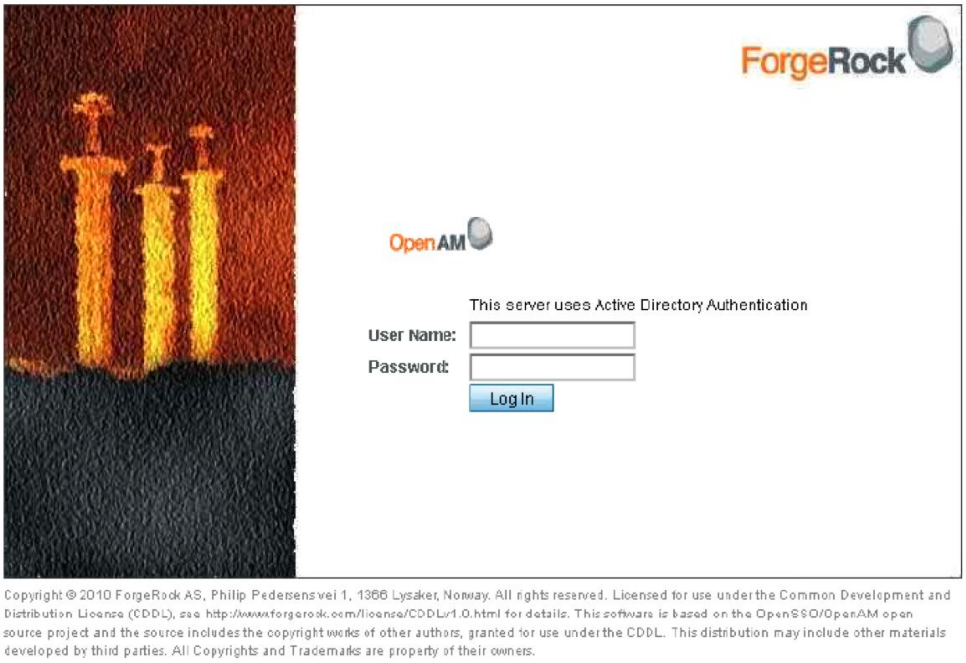
1. Log in to vCloud Director which redirects to the NGC client login.



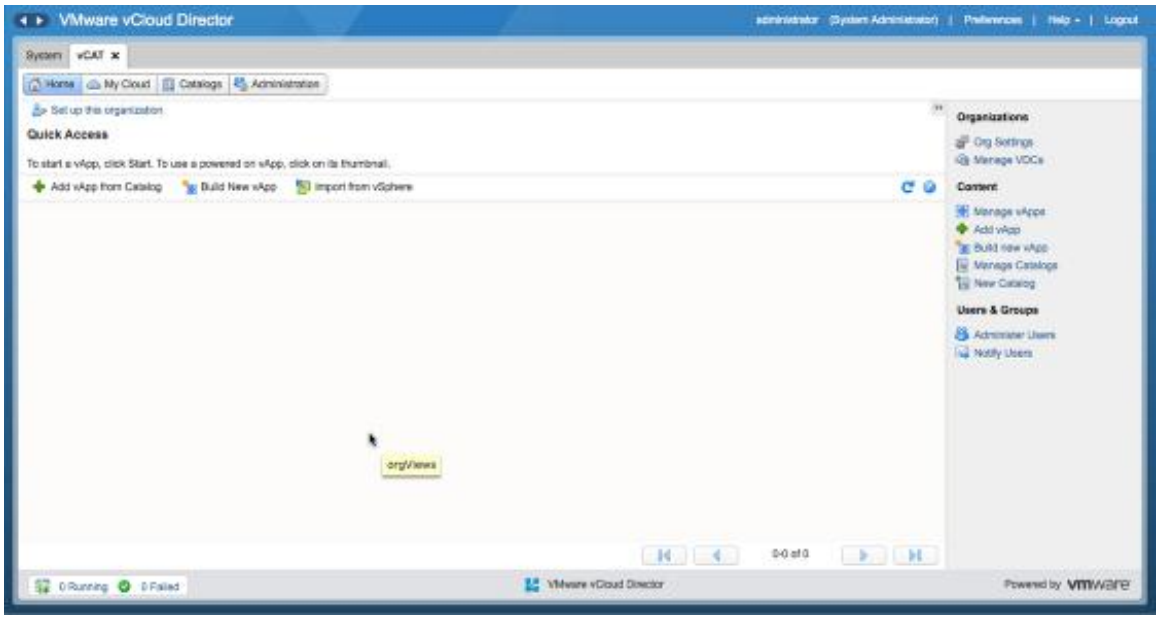
2. The login and authentication takes place on the NGC.



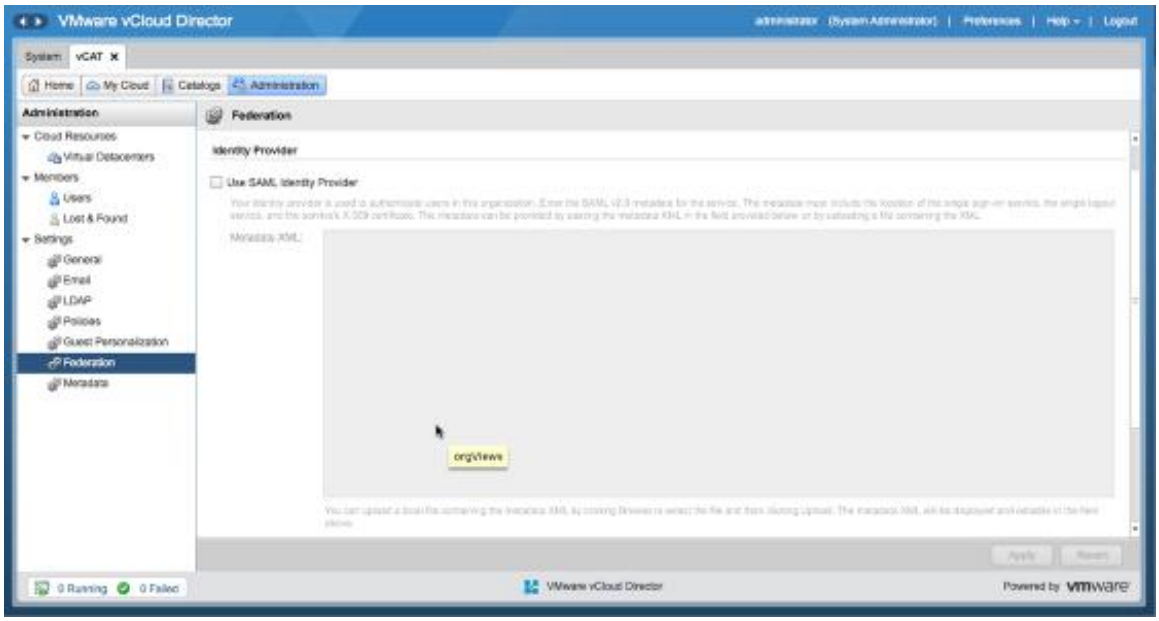
3. Organization Scope: vCloud Director redirects to your IdP, which is OpenAM in this example.



4. Organization Scope: OpenAM redirects back to vCloud Director, using the vCAT Organization in this example.

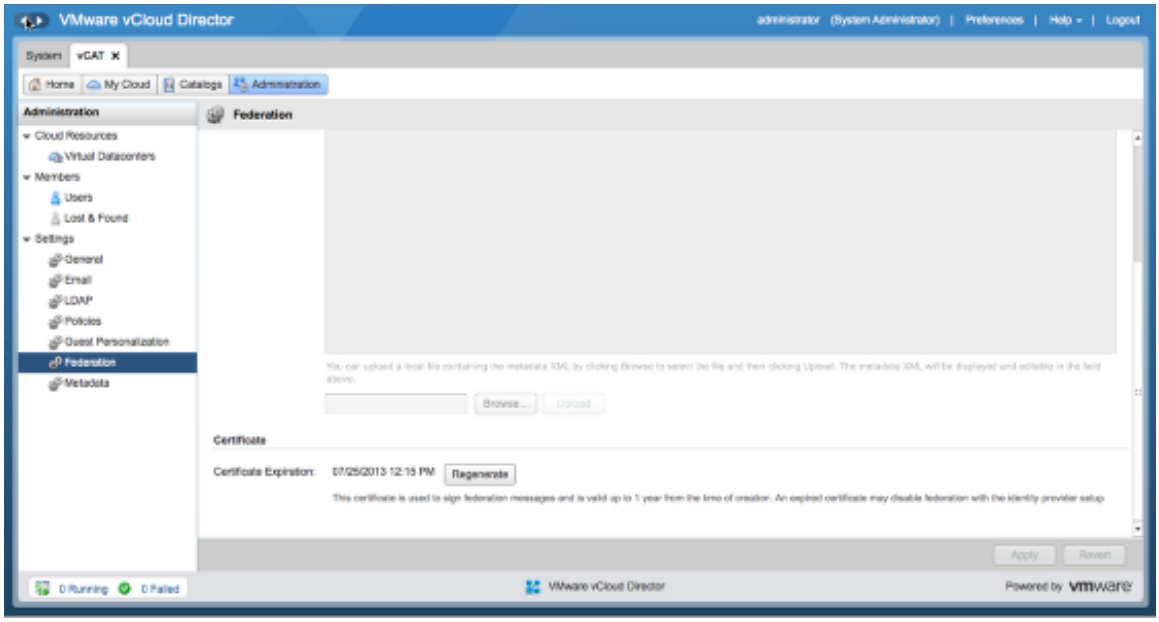


5. Perform installation and configuration of the vCAT Organization Scope with a third-party IdP, which is OpenAM in this example.



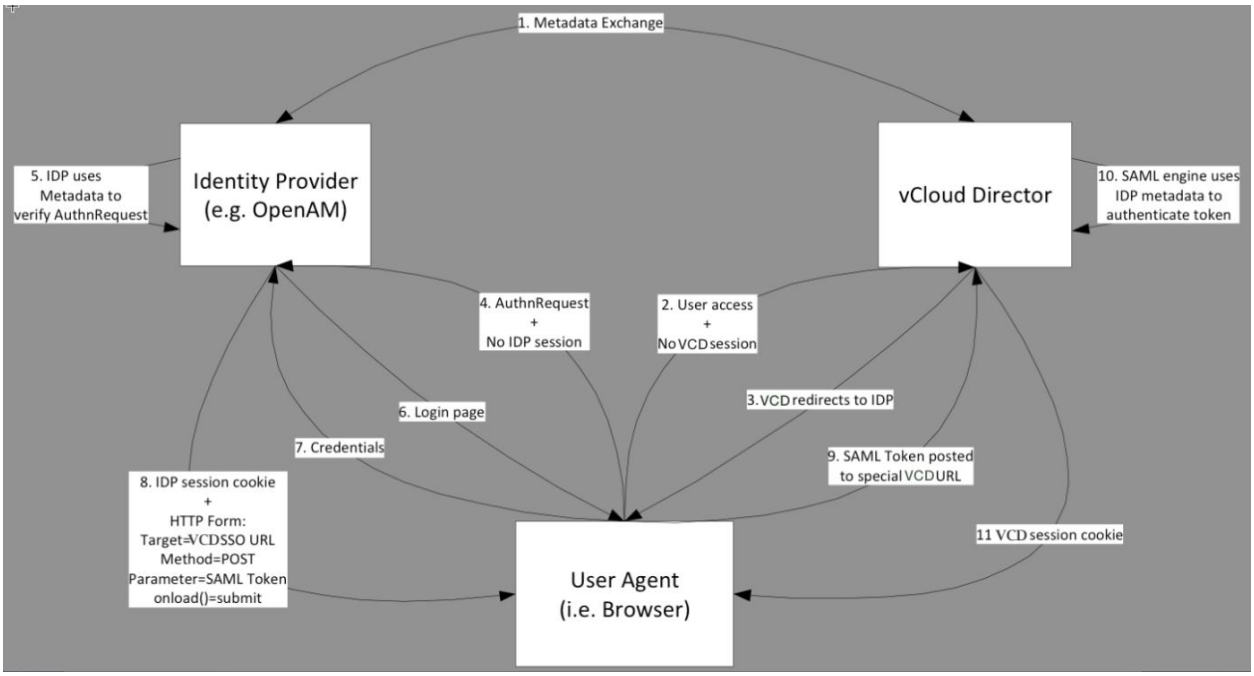
6. Apply the **Use SAML Identity Provider** and paste the XML that you have copied from OpenAM, and **Apply** the changes.

   A **Certificate Expiration** option is also displayed. For now, you can ignore the certificate generation, as this is only required if your certificate is about to expire (in one year).



The following figure illustrates these sequential steps.

Figure 43. Consumer Workflow Detail



7.2.5 Design Implications

- Use single sign-on (SSO) to provide a common service, both internally and externally.
- You must use a supported IdP from VMware.
- Make sure that the SAML assertion contains attributes that vCloud Director understands.
- Make sure that vCloud Director and the IdP are time synchronized to within a few seconds.
- Make sure that vCloud Director and the IdP have valid endpoint certificates.
- Use consistent hostnames or IP addresses while registering with the LookupService.

8. vCloud Management and Monitoring Examples

8.1 vCenter Operations Manager

Deployment Models: private.

Example Components: vCloud Director 1.5, vSphere 5.0, vCenter Operations Manager Enterprise 5.0.1, vFabric Hyperic 4.6.5.

8.1.1 Background

Understanding and managing the unique operational challenges of the cloud is key to success for any cloud provider including private and public. VMware vCenter Operations Manager and VMware vFabric Hyperic® can be used proactively to monitor both the service provider cloud resources and management environment. The cloud resources are the VMware clusters that provide the compute, storage, and network resources to the customers or consumers. The management environment handles the cloud management components.

Using vCenter Operations Manager, the service provider can monitor the resource cluster for overall health and capacity. Health information includes metrics for CPU, memory, disk and network workloads, events, and anomalies. Capacity reflects the resources and capacity available for future client deployments.

Using Hyperic, the service provider can monitor the management components down to the application level. This includes SQL, vCenter, and VCD database and operating system-specific metrics.

8.1.2 Use Case

A vCloud-powered service provider has deployed two vSphere HA/DRS clusters. The first cluster hosts their management stack and the second cluster hosts the customer's actual cloud workloads and maps directly to a provider virtual datacenter. The service provider provides certain SLAs and wants to have a dashboard interface to display the data.

The service provider needs custom alerting in the case that certain SLAs are not met.

The service provider must monitor the following key metrics related to a virtual environment:

- Resource cluster
  - CPU, memory, storage, and network
  - External network switches
  - Capacity (consumed and remaining)
- Management cluster
  - Specific SQL databases (vCenter and VCD)
  - VCD cell transfer disk space

8.1.3 vCenter Operations Manager Example

vCenter Operations Manager can be configured for different types of workload scenarios. For this scenario, vCenter Operations Manager is deployed as a medium type deployment which is appropriate for supporting an environment of approximately 3,000 virtual machines.

The vCenter Operations Manager vApp consists of two virtual machines, one for the UI and the other for analytics, as shown in Figure 44.

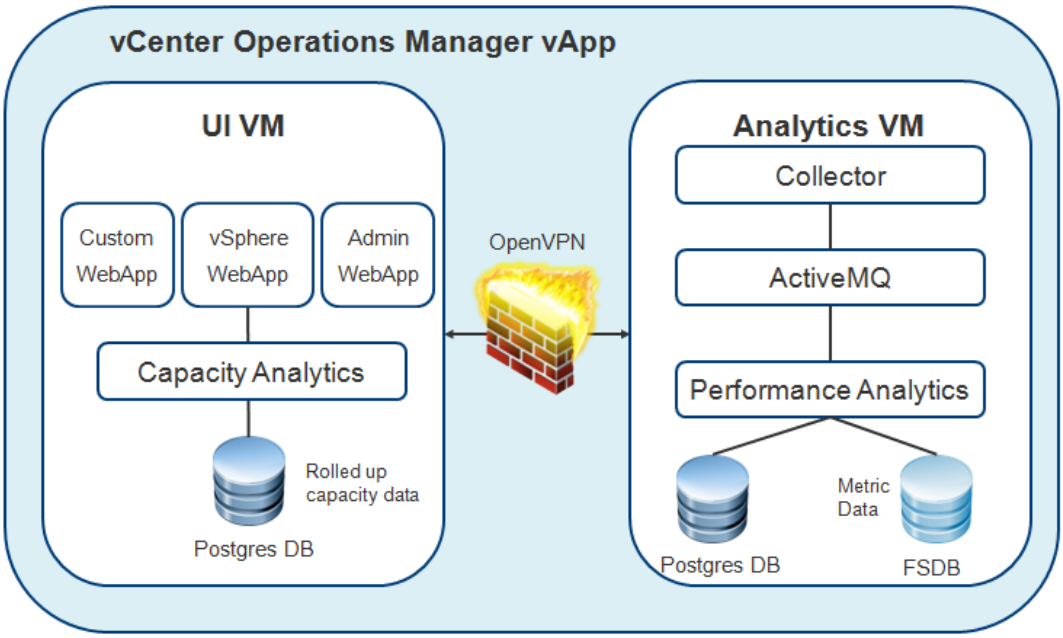
The UI VM is responsible for the following:

- vSphere web application
- Custom web application
- Administration web application

The Analytics virtual machine is responsible for collecting the data from the endpoints. In this example, data is collected from vCenter and Hyperic. Both of these endpoints have software adapters installed that will be configured to collect data. The analytics virtual machine is responsible for the following:

- Capacity and performance analytics
- Capacity collector
- File system database
- PostgeSQL database

Figure 44. vCenter Operations Manager vApp Components



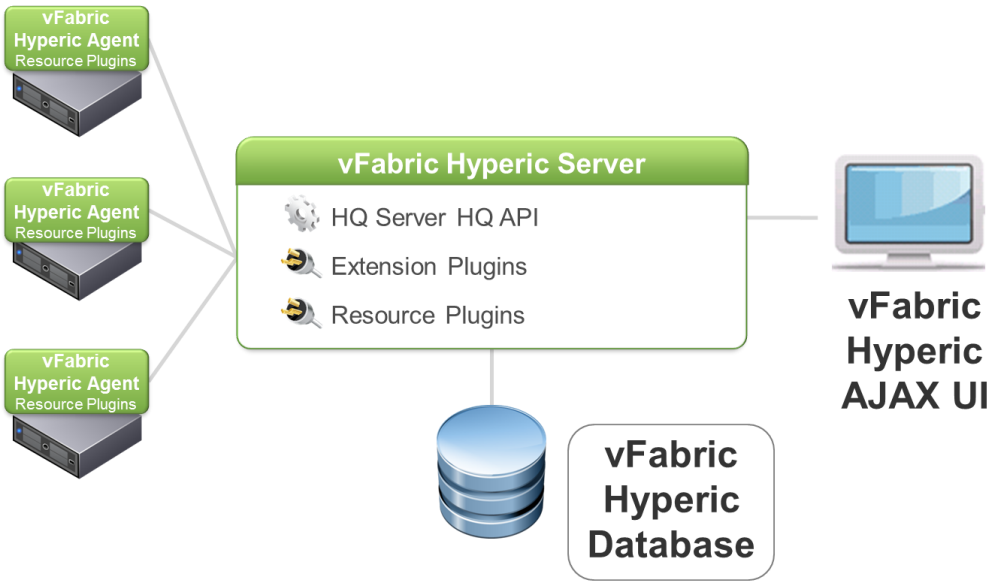
8.1.3.1. vFabric Hyperic

The Hyperic component in this example is used primarily as a relay of data to the vCenter Operations Manager system from individual key vCloud infrastructure servers in the Management cluster. Hyperic is used for obtaining metrics and information about services and processes that are not reported through the vCenter integration with vCenter Operations Manager. Although vCenter can report virtual and physical hardware usage, it is Hyperic that provides granular metrics for the operating systems and their various software components.

For example, Hyperic provides metrics relating to the status of the processes and services that are critical to the functioning of the vCloud Director cells.

The main components of VMware Hyperic are shown in the following figure and include Hyperic Server, agent, database, and the Hyperic user interface, also known as the Hyperic Portal.

Figure 45. Hyperic Configuration



8.1.3.2. vFabric Hyperic Agent

A Hyperic agent is installed on each physical or virtual machine that you want to manage with Hyperic. Agents auto-discover the software components running on the machine, and periodically rescan the platform for changes in its configuration. Hyperic agents gather performance and availability metrics; perform log and event tracking, and allow you to perform control functions such as starting and stopping servers. Agents send the inventory and performance data they collect to the Hyperic server.

8.1.3.3. vFabric Hyperic Server and vFabric Hyperic Database

The Hyperic server receives inventory and metric data from the Hyperic agents and stores that data in the Hyperic database. The server provides facilities for managing your software inventory. It implements the Hyperic inventory and access model, which allows you to group your software assets in useful ways that ease the process of monitoring and management. The Hyperic server detects when alerts fire, and performs the notifications or escalation processes you define. It also processes actions that you initiate through the Hyperic portal or Hyperic web services API. Hyperic server also provides authentication services, using an internal engine or an external authentication service.

8.1.3.4. vFabric Hyperic User Interface (Hyperic Portal)

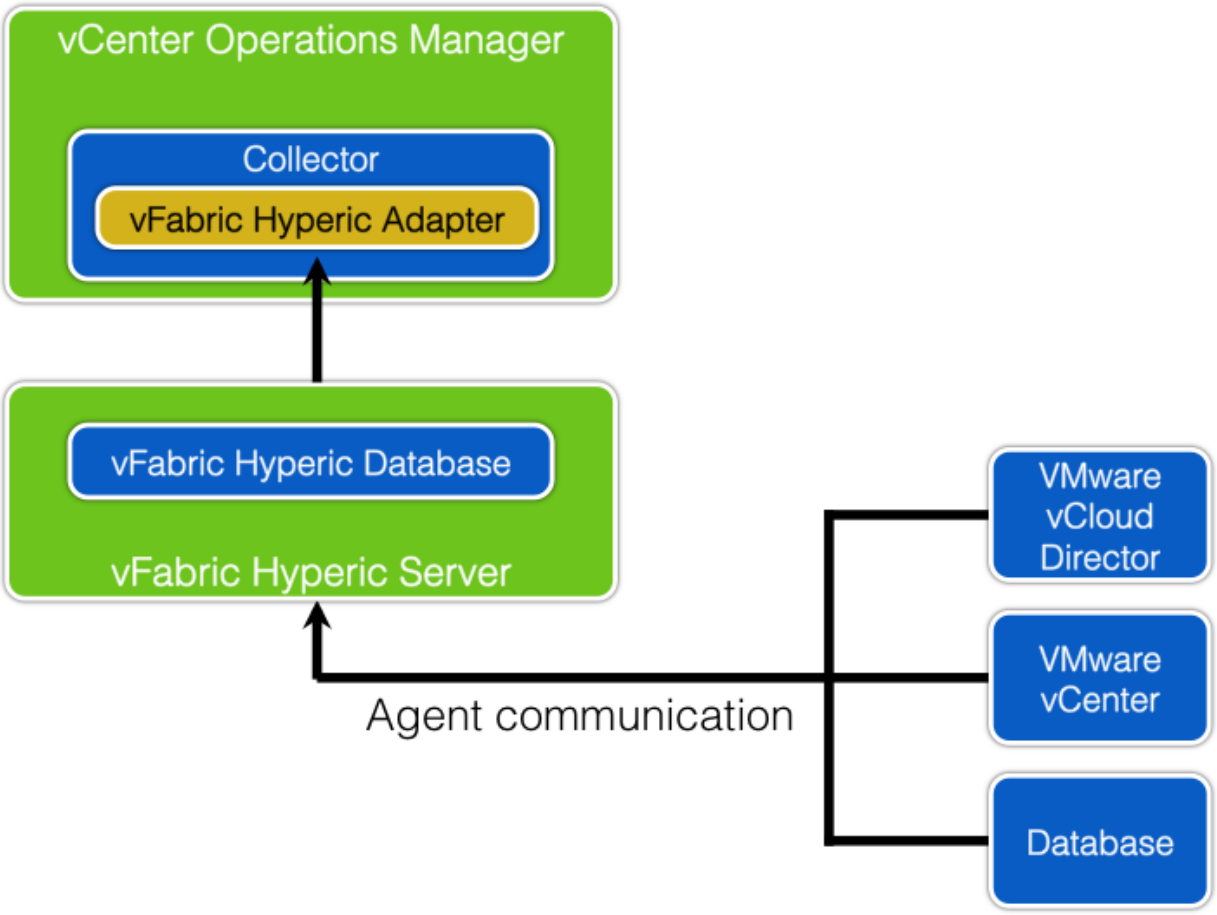
The Hyperic browser-based user interface, sometimes referred to as the *Hyperic Portal* is a configurable, extendable user interface for monitoring and analyzing performance and availability.

Refer to the *VMware vFabric Hyperic Documentation* for additional information about using vFabric Hyperic (http://www.vmware.com/support/pubs/vfabric-hyperic.html).

8.1.4 vCenter Operations Manager and vFabric Hyperic Integration

As detailed in the previous section, Hyperic is used as a source of data for vCenter Operations Manager from individual key vCloud infrastructure management servers. This proxy functions as a result of the vCenter Operations Manager Hyperic adapter that is installed and enables the required process and service information to be presented to vCenter Operations Manager.

Figure 46. vCenter Operations Manager and Hyperic Integration



As shown in the figure, vCenter Operations Manager Enterprise supports the Hyperic Adapter which enables the exchange of information between vCenter Operations Manager Enterprise and Hyperic HQ data. The adapter's functionality includes the following:

- Connects to and collects data from Hyperic HQ.
- Transforms the collected data into the format vCenter Operations Enterprise is designed to consume.
- Passes the data to the vCenter Operations Enterprise collector for final processing.

In addition to the standard adapter functions, the adapter supports manual and auto discovery and the ability to create relationships based on the associations found in Hyperic HQ.

8.1.4.1. How Data Is Retrieved

The Hyperic adapter's data source is the Hyperic HQ database. The adapter uses standard JDBC access to the database for running SQL queries. The adapter uses the vCenter Operations API to communicate and deliver data to vCenter Operations.

8.1.4.2. Retrieved Data

The vCenter Operations Manager Hyperic adapter can collect data from different resources and creates resource kinds dynamically according to platform types and server types in Hyperic. Resources in vCenter Operations represent Hyperic platforms and servers.

Resources in Hyperic are one of the following kinds.

- A platform, which is either an operating system platform such as Linux, Solaris, Windows 32-bit, or UNIX, or a virtual and network platform such as Cisco IOS, GemFire Distributed System, NetApp Filer, vSphere Host, or a vSphere virtual machine.
- A server, which is a software product that runs on a platform, such as Tomcat, JBoss, Exchange, SQL, or Oracle.
- Services and platform services, which are software components that run on either a server or a platform. Server services can be either internal server components such as database tables or .NET applications or they can be a deployed item such as *CustomerEntityEJB*. Examples of platform services include DHCP, DNS, and CPU and network interfaces.

The Hyperic adapter can also collect data for Hyperic services. Service metrics are stored under a specific instance and group and the service name is used as the instance name.

Sources map to vCenter Operations resources using resource names in vCenter Operations to match the names from the `eam_resource` table in the Hyperic HQ database. All available types of resources in the source system can be collected.

The adapter can import relationships between Hyperic platforms and servers. Relationships are based on the `platform_id` column value in the `eam_server` table. Relationships are imported during manual discovery only.

8.1.4.3. Metrics

The adapter can collect all metrics for Hyperic platforms, servers, and services. Use a single query for each adapter instance to retrieve the metric data for children instances. Metrics are collected using a query to the `HQ_METRIC_DATA` tables and viewed using a time filter.

- Availability metrics are collected from the `HQ_AVAIL_DATA_RLE` table. The system current time is used as the timestamp of the collected Availability metric.
- Collection interval used is assigned to adapter instance resource and attribute package.

8.1.4.4. Events

There are no events collected from Hyperic HQ.

8.1.4.5. Prerequisites

The adapter supports Hyperic HQ versions 3.x and 4.0.

8.1.4.6. Database Connection Configuration

The following table lists the default vCenter Operations Enterprise port connections for supported databases. You must specify the port number when you configure the adapter instance resource.

Table 15. vCenter Operations Enterprise Port Access Requirements

Database	Port Number
MySQL	3306 TCP
Oracle	1521 TCP

The following table lists the default vCenter Operations Enterprise URLs used to connect to data sources. The Hyperic adapter requires a standard JDBC connection URL.

Table 16. vCenter Operations Manager Data Source URLs

Database	URL
MySQL	jdbc:mysql://<db_host>:<db_port>/<hqdb_name>
Oracle	jdbc:oracle://<db_host>:<db_port>/<hqdb_name>

8.1.5 vCenter Operations Manager Dashboards

To get the full value from vCenter Operations Manager for monitoring a vCloud environment, it is recommended to configure vCenter Operations Manager dashboards. These dashboards provides a view of the health of the various vCloud constructs.

Dashboards can be shared between Admin groups. An example of this is the disk and network dashboards. The storage administrators can have a dashboard that is related only to storage metrics from the cluster. This dashboard can include metrics such as cluster disk I/Os or read/write latency. At the same time the network administrators have a dashboard that is related to the cluster networking metric. These metrics can include physical switches that are connected to the vSphere cluster to give the network administrator insight to statistics from the virtual and physical environment on one dashboard.

For this example, a dashboard has been created to display the following statistics on the resource cluster:

- Capacity remaining.
- Alerts and events.
- Network.
- Storage.
- Memory.
- CPU.

A second dashboard gives statistics on the management cluster. The dashboard has been configured with the following metrics:

- vCenter SQL database transactions/database size.
- vCloud Director SQL database transactions/database size.
- SQL Server operating system drive space remaining.
- vCenter Server operating system drive space remaining.
- vCloud Director mount point space remaining.

8.1.5.1. vCenter Operations Manager Widget Configuration

The custom UI of vCenter Operations Manager uses widgets to display information about objects that are being monitored. Some widgets display only data and other can be configured to display data and have thresholds to change color when thresholds are exceeded. Following are two examples of how to configure widgets.

8.1.5.2. Generic Scoreboard

The widget configuration settings are shown in the following table.

Setting	Value
Widget Title	Displayed title of widget
Self Provider	On
Refresh Widget Content	On
Widget Refresh Interval	300 (seconds)

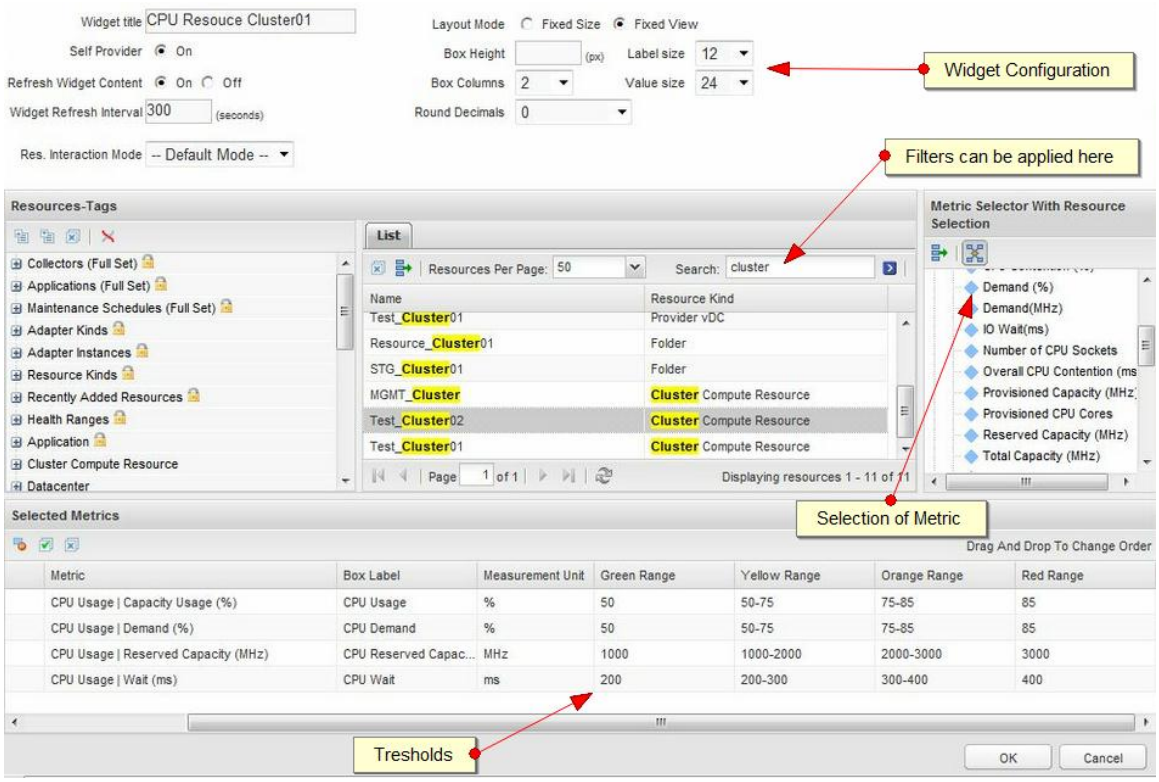
After the widget settings have been set, the relevant objects and metrics must be selected. This example uses a filter for *cluster*. Then the cluster for which we want to display statistics, and metrics, are selected. After all the desired metrics are listed in Selected Metrics, the thresholds can be configured.

The following table shows the thresholds that can be set.

Threshold	Range
Green	Up to 10
Yellow	10–20
Orange	20–30
Red	30 and higher

The following figure shows the completed widget.

Figure 47. Generic Cluster CPU Scoreboard



8.1.5.3. Heat Map

The Heat Map widget can be used when a few objects (for example, datastores, cluster physical CPU cores) must be displayed in comparison with each other. As an example, physical CPU cores can be displayed for all hosts in a cluster, and those over a certain threshold are displayed as red to identify hot spots.

The widget configuration settings are shown in the following table.

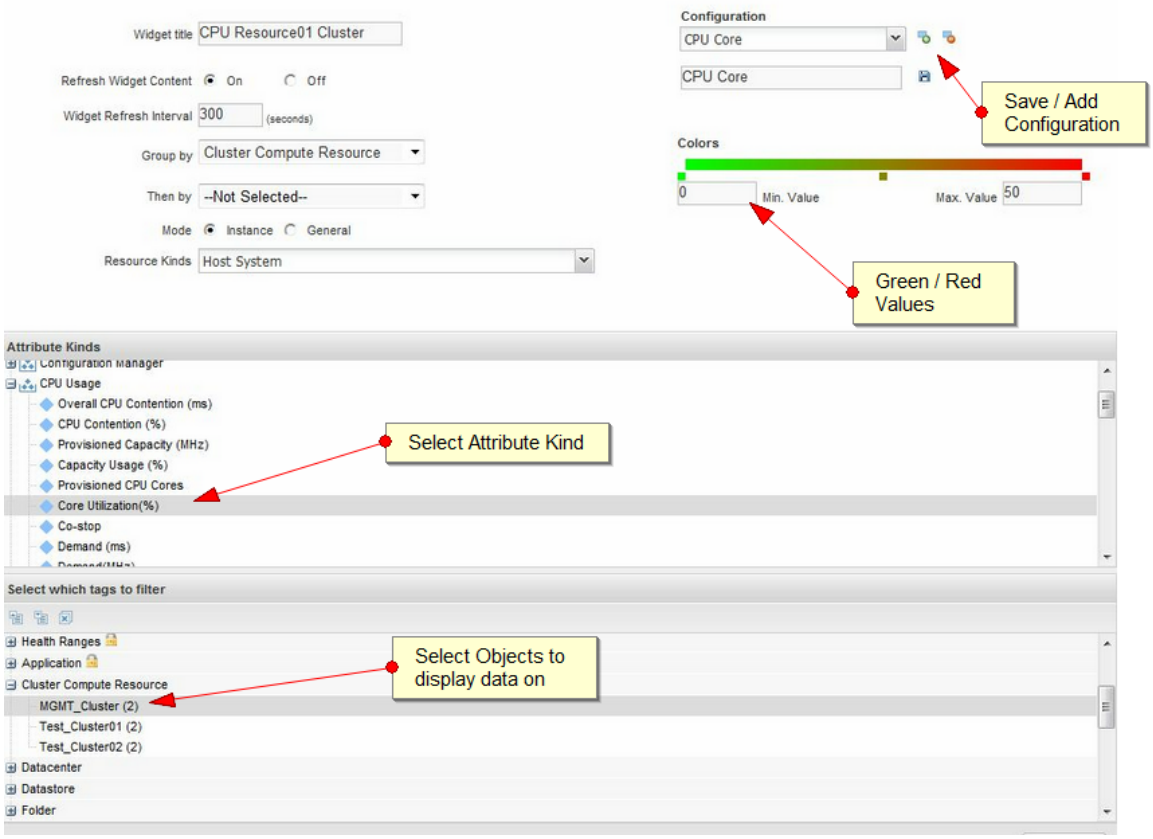
Setting	Value
Widget Title	Displayed title of widget
Self Provider	On
Refresh Widget Content	On
Widget Refresh Interval	300 (data collection is every 300 seconds)
Group By	Select object to group by
Resource Kind	Heat map data displayed for which type of resource

After the resource kind has been selected, choose the reported metric by selecting the attribute kind. Then select the tag from which the data will be reported.

After the selection of metric and tags, the configuration must be saved. Click the green plus sign to give the configuration a name and save it. Multiple configurations can be saved and then selected to be displayed on the same widget.

The completed heat map widget is shown in the following figure.

Figure 48. Management Cluster CPU Core Utilization Heat Map

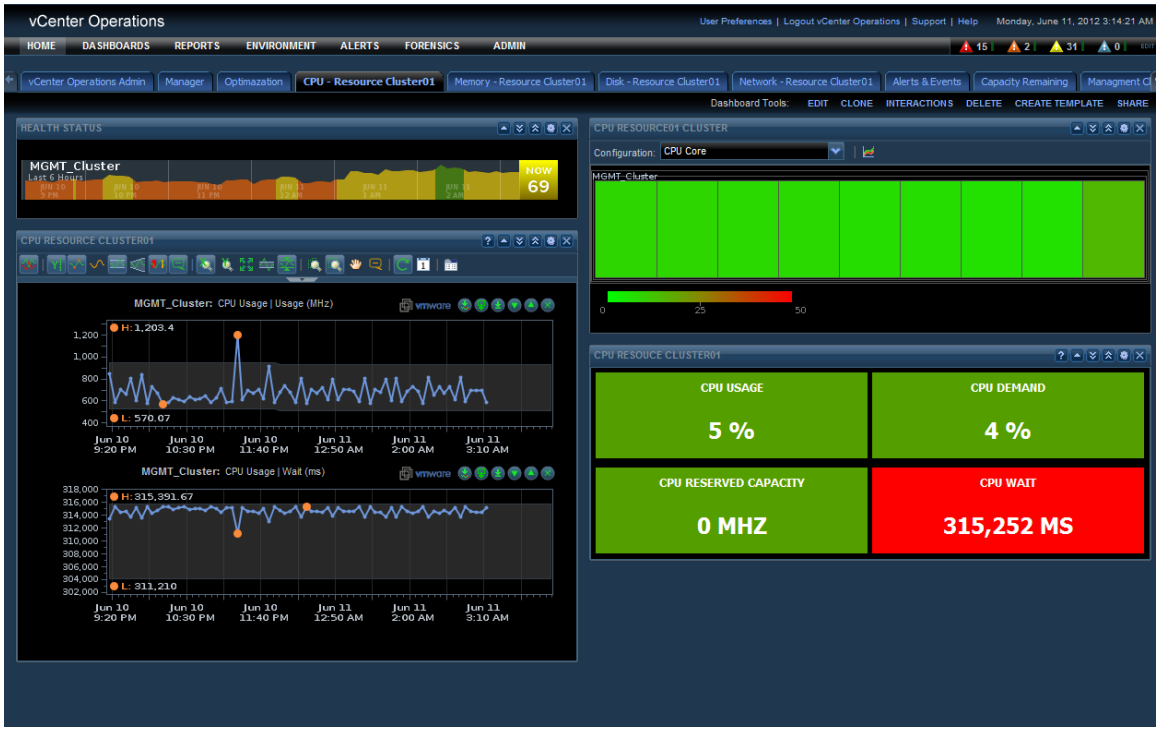


8.1.5.4. Physical CPU Resource Monitoring of Resource Clusters

The CPU Dashboard can be used to monitor the physical CPU performance of the resource cluster. Figure 49 shows a sample dashboard that has been created using four widgets, which are illustrated in Figure 50, Figure 51, Figure 52, and Figure 53. These widgets include:

- Generic Scoreboard (Figure 50).
- Heat Map (Figure 51).
- Metric Graph (Figure 52).
- Health Status (Figure 53).

Figure 49. CPU Resource Cluster Dashboard



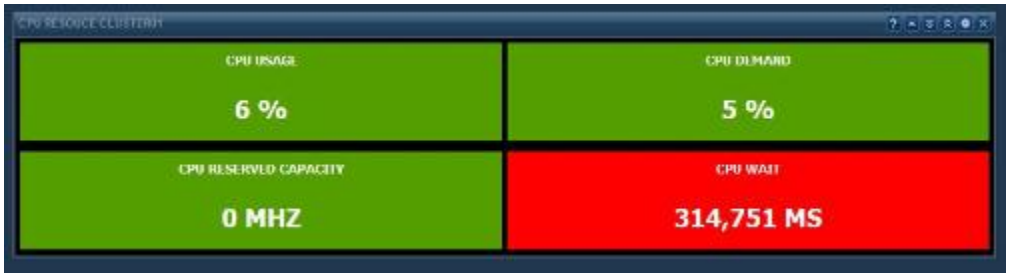
The Generic Scoreboard widget is at the bottom right of the dashboard in Figure 49. The selection is based on the resource cluster metric which is a sum of all host metrics in the cluster.

Table 17. Generic Scoreboard Widget Configuration

Metric	Unit	Green Range	Yellow Range	Orange Range	Red Range
Capacity Usage %	%	50	50–75	75–85	85
Demand	%	50	50–75	75085	85
CPU Reserved Capacity	MHz	100000	100000–150000	150000–175000	175000
Wait	ms	100	100–200	200–300	300

These threshold/ranges are only examples. These values should be based on the cluster design threshold values and the customer requirements.

Figure 50. Resource Cluster CPU Scoreboard Widget

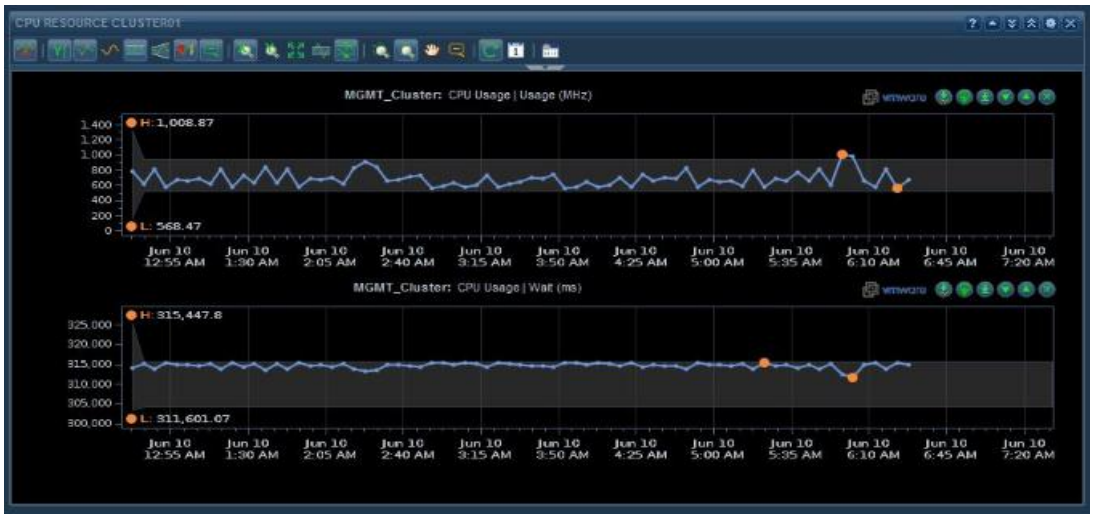


The Heat Map widget shows the resource cluster physical CPU core utilization. This displays all the ESXi physical CPUs and cores in the resource cluster to identify hot spots on the physical CPUs. This configuration uses the core utilization metric and resource cluster tag to display the heat map data.

Figure 51. Resource Cluster Physical CPU Core Heat Map Widget

The Metric Graph widget is used to give a graph view of some of the CPU metrics. The view is customizable to display from *last hour* to *last year*. The graph can also display the dynamic thresholds for certain metrics. As an example, this widget displays the cluster CPU usage and cluster CPU wait.

Figure 52. Resource Cluster CPU Metric Graph Widget



The Health widget is used to display overall Resource Cluster Health. The widget can be configured to display data from the *last hour* to *last month*.

Figure 53. Cluster Health Widget

8.1.5.5. Memory Dashboard

The Memory Dashboard can be used to monitor the cluster memory usage and demand. Figure 54 shows a sample dashboard that has been created using four widgets. These widgets include:

- Generic Scoreboard (Figure 55).
- Heat Map (Figure 56).
- Metric Graph (Figure 57).
- Health Status (Figure 53).

Figure 54. Cluster Memory Dashboard

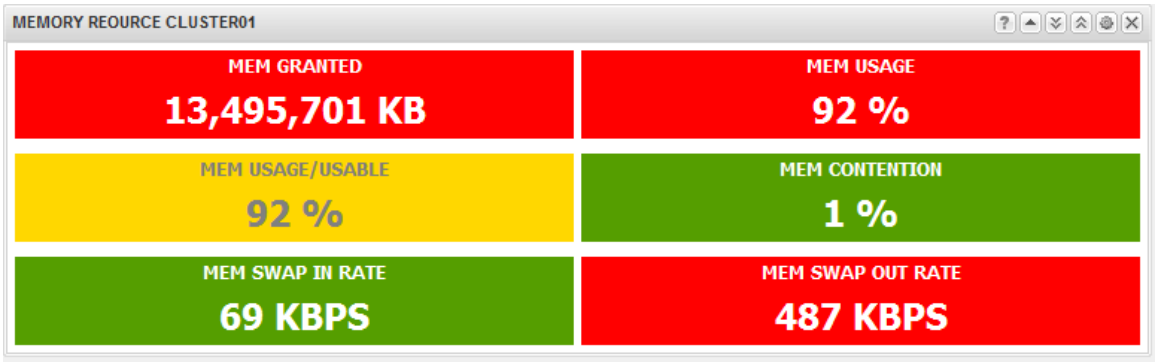
The Generic Scoreboard widget has been configured with these metrics defined at the cluster level. The Resource cluster is used for the memory metric selections.

Table 3: Generic Scoreboard Configuration

Metric	Unit	Green Range	Yellow Range	Orange Range	Red Range
Memory Granted	KB	8000	8000–12000	12000–13000	13000
Usage	%	50	50–75	75–80	80
Usage/Usable	%	50	50–75	75–80	80
Contention	%	50	50–75	75–80	80
Swap In Rate	KBps	300	300–400	400–500	500
Swap Out Rate	KBps	5	5–10	10–20	20

Note: These threshold/ranges are only examples. These values should be based on the cluster design threshold values and the customer requirements.

Figure 55. Cluster Memory Scoreboard Widget



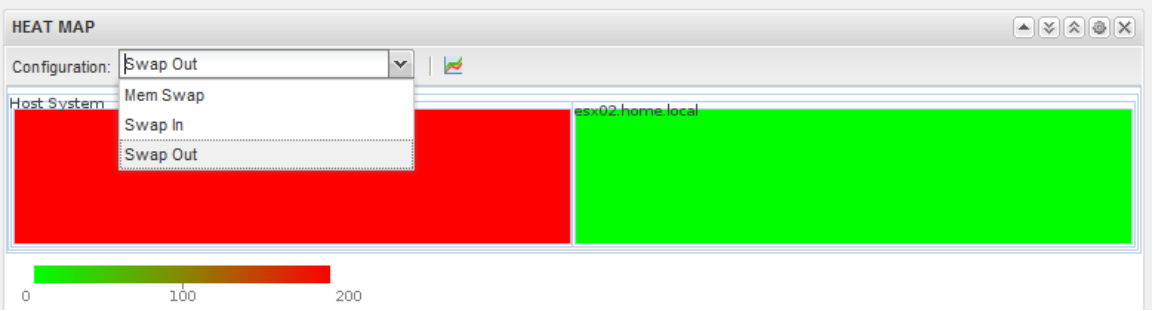
The Heat Map is configured with three metrics, created from the cluster resource, as follows:

- Swap Out Rate (KBps).
- Swap In Rate (KBps).
- Mem Swap Used (Kb).

The settings for the Heat Map widget are shown in the following table.

Metric	Value
Attribute Kinds	Memory
Tags to Filter	Cluster compute resource (select the correct cluster)
Minimum Value	(Custom setting to environment)
Maximum Value	(Custom setting to environment)

Figure 56. Cluster Memory Heat Map Widget

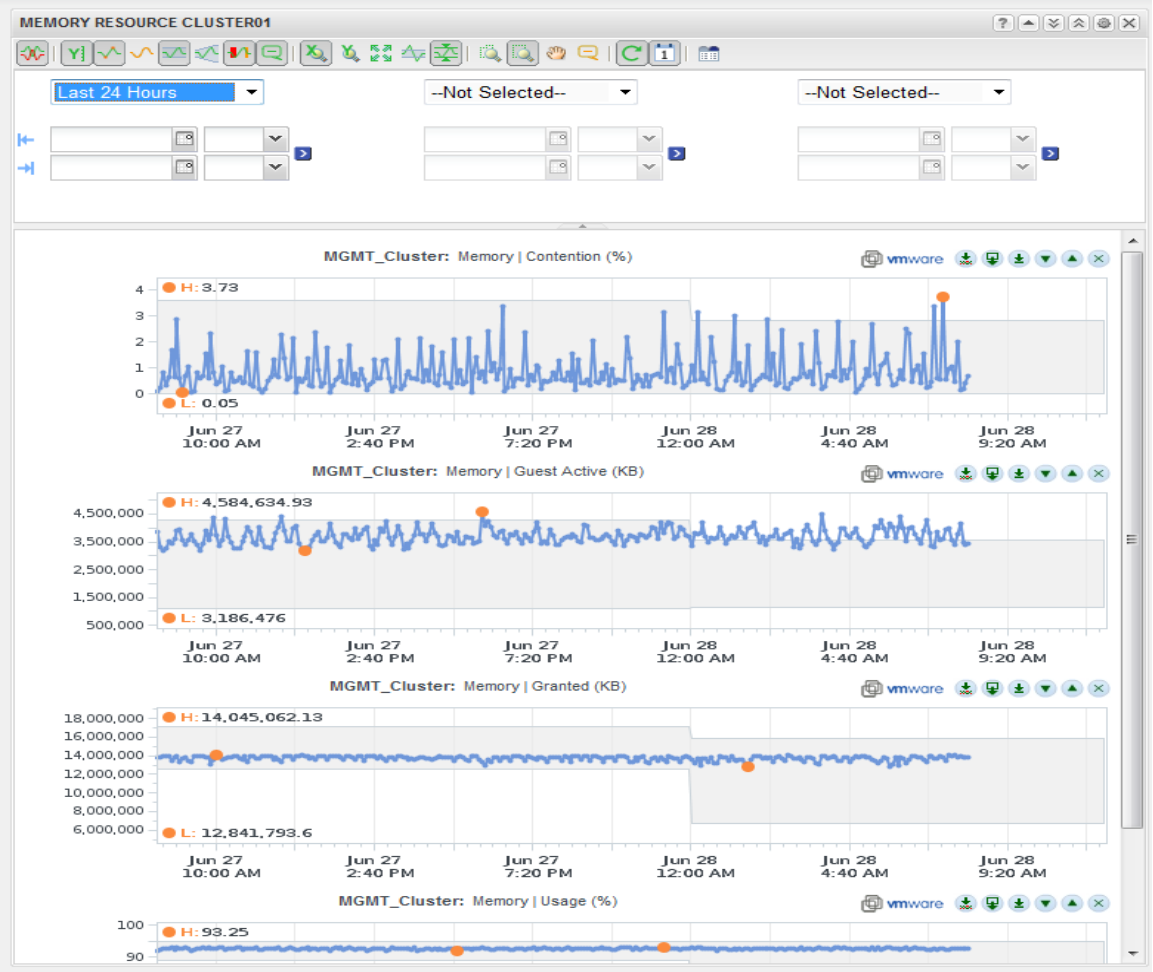


The Metric Graph Widget can display line graphs of historical memory statistics. This example selects the following memory metrics to be displayed over a 24-hour period. Note that the historical time can be changed to up to one year and can also indicate the DT (dynamic threshold) values.

Table 18. Cluster Memory Metric Graph Settings

Metric
Memory | Contention (%)
Memory | Guest Active (KB)
Memory | Granted (KB)
Memory | Usage (%)

Figure 57. Cluster Memory Metric Graph Widget



8.1.5.6. Storage Dashboard

The Storage Dashboard widget can help with troubleshooting at the cluster or host level and can be used by the virtual infrastructure admins and the storage administrators. Figure 58 shows this dashboard which is made using four widgets:

- Disk Scoreboard (Figure 59).
- Disk Capacity Scoreboard (Figure 60).
- Cluster Storage Metric Graph (Figure 61).
- Health Status (Figure 53).

This dashboard focuses on cluster statistics and datastore hot spots. Hot spots in this example are related to latency to the datastore as detected by the ESXi host. Heat maps can also be used to identify datastores with low capacity remaining, which is done using a super metric.

Figure 58. Storage Dashboard

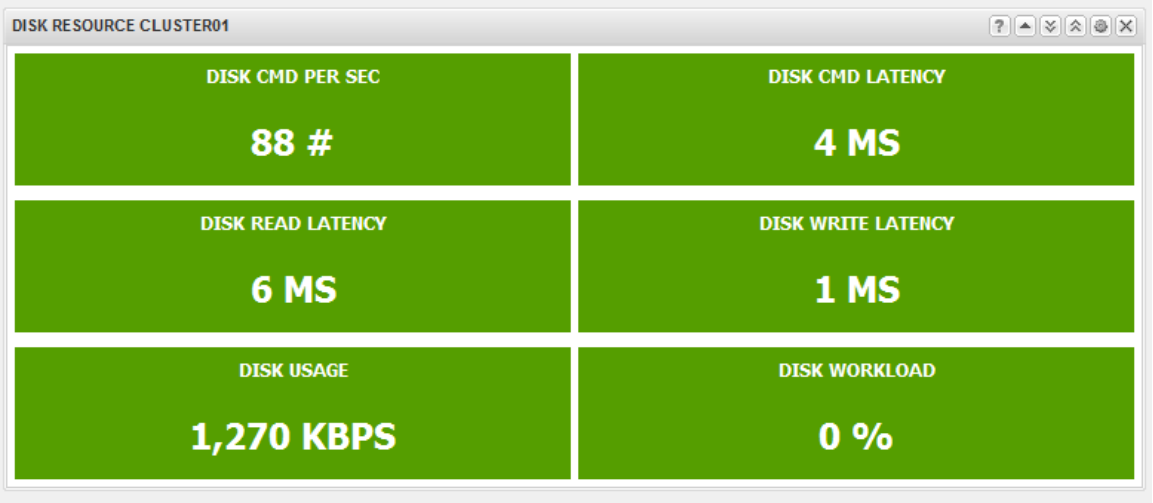
This dashboard uses two Scoreboard widgets. The first widget displays statistics related to the cluster I/O usage. The second widget displays information about virtual memory snapshot space and virtual disk usage.

Table 19. Cluster Storage Usage Widget Settings

Metric	Unit	Green Range	Yellow Range	Orange Range	Red Range
Disk CMD Per Sec	#	200	200–300	300–400	400
Disk CMD Latency	ms	20	20–30	30–40	40
Disk Read Latency	ms	20	20–30	30–40	40
Disk Write Latency	ms	20	20–30	30–40	40
Disk Usage	KBps	10000	1000–2000	2000–3000	3000
Disk Workload	%	50	50–60	60–70	70

Note: These threshold/ranges are only examples. These values should be based on the cluster design threshold values and the customer requirements.

Figure 59. Cluster Disk Scoreboard Widget



The second generic scoreboard widget displays virtual machine disk space information. This example shows how much snapshot space is used and the total amount of virtual disk space used.

Table 20. Cluster Disk Capacity Scoreboard Settings

Metric	Unit	Green Range	Yellow Range	Orange Range	Red Range
Virtual Machine Snapshot Space	GB	1	1–10	10–20	20
Virtual Disk Usage	GB	90	90–200	200–500	500

Note: These threshold/ranges are only examples. These values should be based on the cluster design threshold values and the customer requirements.

Figure 60. Cluster Disk Capacity Scoreboard Widget

The Metric Graph widget displays historical disk statistics that include the DT values. The DT values are also displayed for the next number of hours, depending on the Date Control selection). In Figure xxx, we have enabled the Anomalies view also (all the yellow spikes). For the example we are displaying historical stats for the following metrics:

Metric	Unit
Disk	Commands per second
Disk	Disk commands latency (ms)
Disk	Usage rate (KBps)
Disk	Disk write latency (ms)
Disk	I/O usage capacity

Figure 61. Cluster Storage Metric Graph Widget

8.1.6 Network

The Network Dashboard uses vCenter Operations Manager data collected from vCenter and from Hyperic. The Hyperic server uses SNMP to collect data from a Cisco 3550 switch.

The completed network dashboard widget is displayed in Figure 62, and is made from four widgets:

- Outbound/Inbound Packet Rate Metric Graph (Figure 63).
- Network Scoreboard (Figure 64).
- Physical NIC Heat Map (Figure 65).
- Network Performance Metric Graph (Figure 66).

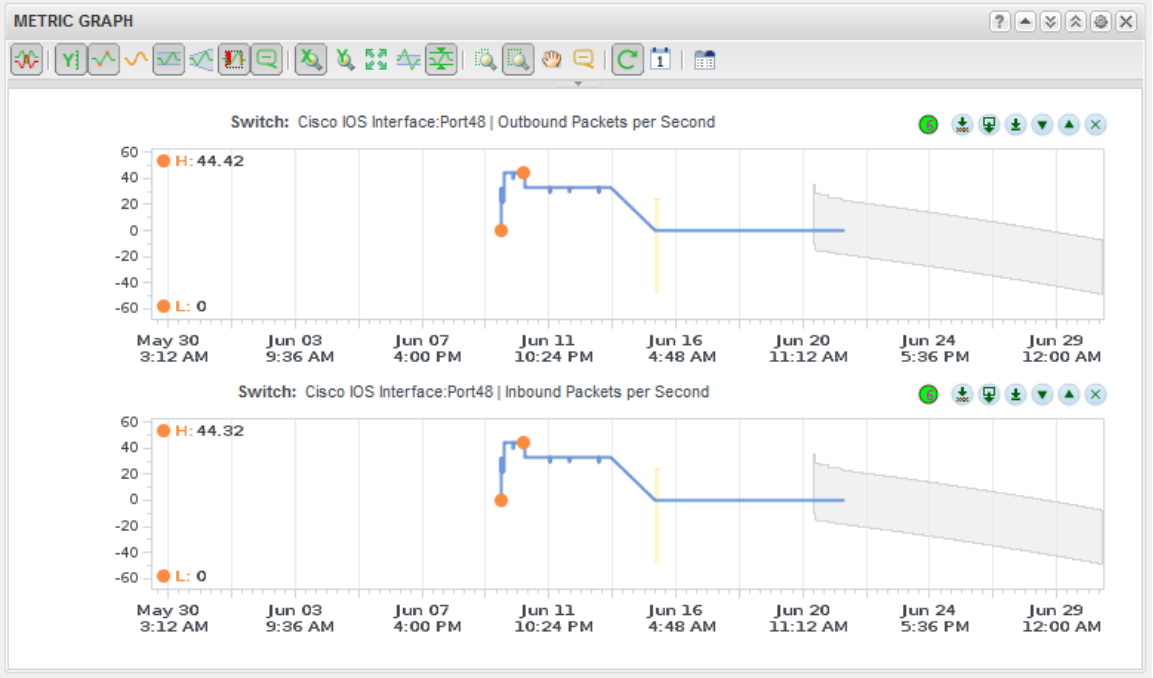
Figure 62. Network Dashboard Widget

This example monitors port 48 on the switch for outbound and inbound packets. The metric graph displays these two metrics.

Metric	Unit
Cisco IOS Interface:Port48	Outbound packets per second
Cisco IOS Interface:Port48	Inbound packets per second

The completed widget appears as shown in Figure 63.

Figure 63. Outbound and Inbound Packet Rate Metric Graph



Using the Scoreboard Widget and some cluster wide statistics, the following widget can be created to show statistics on inbound packets per second network bandwidth usage and errors. This widget uses the following metrics:

Metric	Unit	Green Range	Yellow Range	Orange Range	Red Range
Net Cluster Usage	KBps	100	100–1000	1000–5000	5000
Net Cluster Demand	%	50	50–60	60–70	70
Net Cluster Workload	%	50	50–60	60–70	70
Net Max Throughput	Kbps	6000	6000–6500	6500–7000	7000

Note: These threshold/ranges are only examples. These values should be based on the cluster design threshold values and the customer requirements.

The completed widget appears as shown in Figure 64.

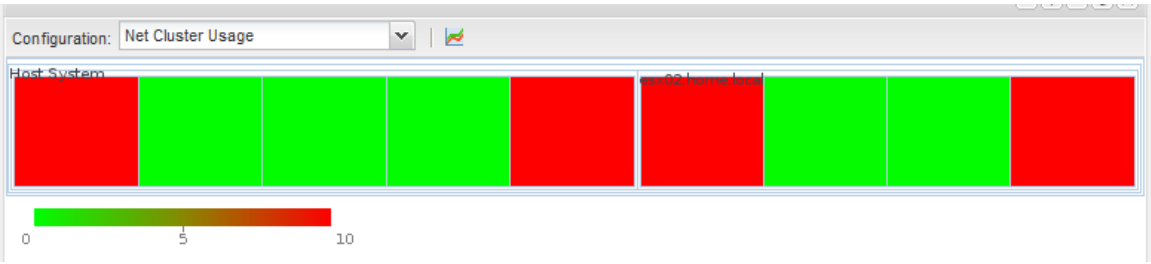
Figure 64. Completed Network Scoreboard Widget

A Heat Map widget can be used to display statistics for physical NIC usage in the ESXi host. The heat map settings for this widget are as shown in the following table.

Metric	Value
Refresh Widget Content	On
Widget Refresh Interval	300
Group by	Resource kind
Mode	Instance
Resource Kinds	Host systems
Attribute Kinds	Network | usage rate (KBps)
Tags to filter	Cluster compute resource | *cluster*

The completed heat map widget appears as shown in Figure 65.

Figure 65. Physical NIC Heat Map Widget

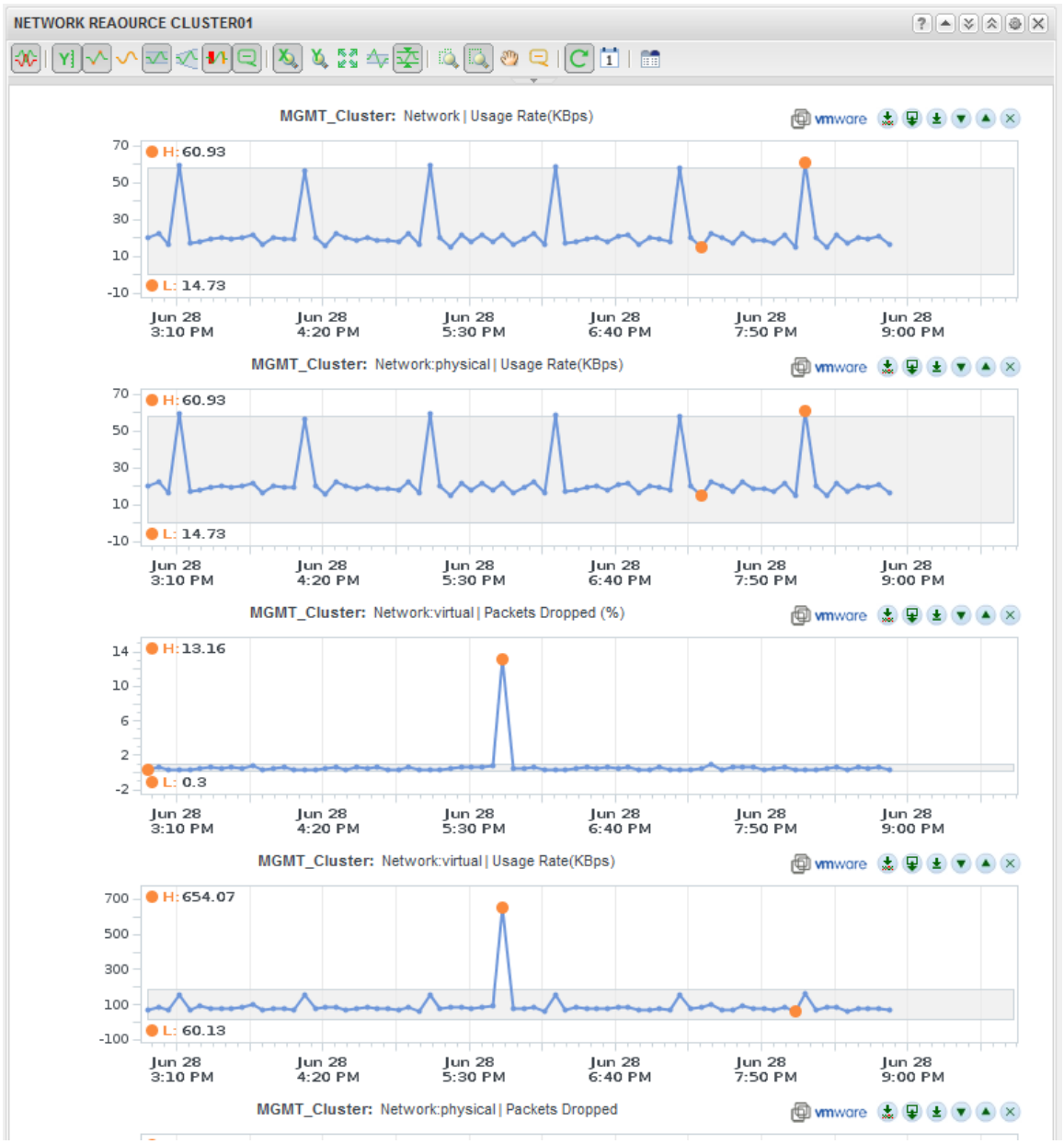


The Metric Graph widget can display historical network details regarding usage and errors. The Widget displays information on both the physical and virtual networking. The widget is configured with the following metrics:

Metric	Units
Network	Usage Rate (KBps)
Network:Physical	Packets Dropped
Network:Physical	Usage Rate (KBps)
Network:Virtual	Packets Dropped (%)
Network:Virtual	Usage Rate (KBps)

The Completed widget appears as shown in Figure 66.

Figure 66. Network Performance Metric Graph Widget



8.1.7 Alerts, Events, and Health

The Health Status Dashboard appears as shown in Figure 67, and is made from the following widgets:

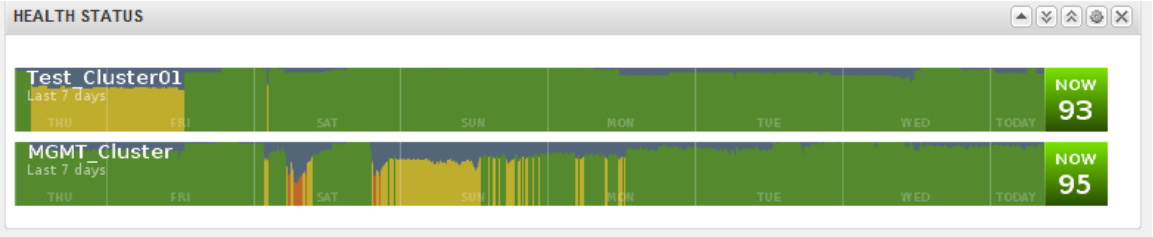
- Cluster Health Bar (Figure 63).
- Alerts (Figure 64).

Figure 67. Health Status Dashboard

The Health widget can be used to display management and resource cluster health information. The widget is configured as follows.

Metric	Value
Self Provider	On
Mode	Self
Refresh Widget	On
Widget Refresh Interval	300
Order by	Health (asc)
Period Length	Last 7 days
Tags to Filter by	Cluster compute resource (select clusters)

Figure 68. Cluster Health Bar



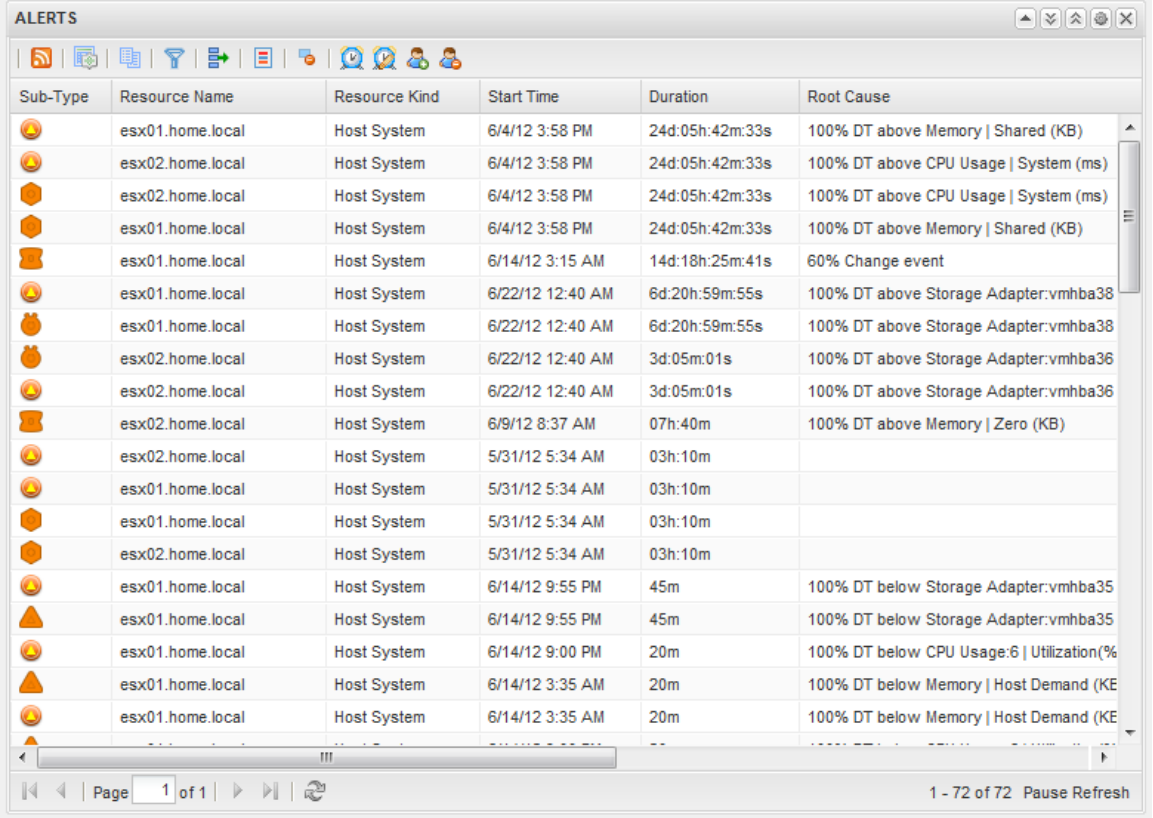
The Alerts Widget can display alerts that have been generated. Alerts in this widget can be displayed for clusters or virtual machines. The correct filter must be applied. This widget displays alerts for the clusters. If needed, the *VM Entity Status* filter can be selected to include *Powered:On* or *Powered:Off* virtual machines.

This widget is configured as shown in the following table.

Metric	Value
Refresh Widget	On
Widget Refresh Interval	300
Tags to Filter	Cluster compute resource (select clusters)

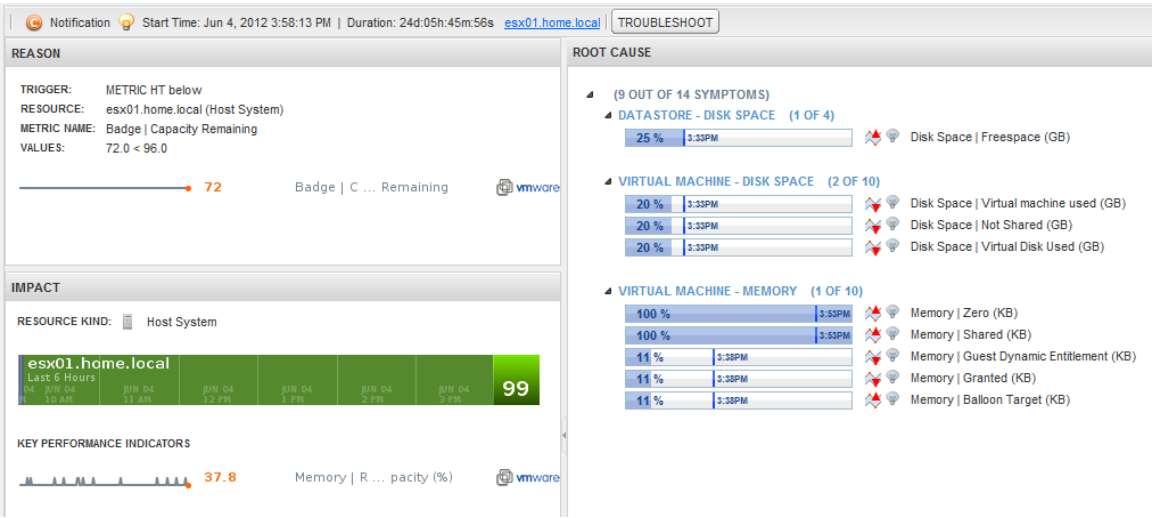
The completed widget displays as follows:

Figure 69. Alerts Widget



Double-click an alert to direct vCenter Operations Manager to open a more detailed page displaying information about the event. Figure 70 displays an example of the details for an alert.

Figure 70. Alerts Information Detail



8.1.8 Capacity Remaining

Capacity remaining in a cluster is useful in determining available capacity for future virtual machine deployments. Public service providers provide the illusion of infinite capacity so they must be able to proactively add capacity as needed on demand. As an example, this information can be used to gauge how many more virtual machines can be deployed on a resource cluster based on the observed average workload profile in the environment. This example looks at the time left before running out of storage, network, and compute resources in a cluster.

The Scoreboard widget shown in Figure 71 is used to display information about available time (days) left in the cluster for CPU, memory, disk, and network.

This example uses the Capacity Remaining metric. The Green threshold is high, and the Red threshold is low. The metric resource selections used in the widget are as follows:

Metric	Unit	Green Range	Yellow Range	Orange Range	Red Range
Count | VM	Days	90	60–90	30–60	30
Count | VM | Disk Space	Days	90	60–90	30–60	30
Count | VM | Disk I/O	Days	90	60–90	30–60	30
Count | VM | Memory	Days	90	60–90	30–60	30
Count | VM | Network	Days	90	60–90	30–60	30

Note: These threshold/ranges are only examples. These values should be based on the cluster design threshold values and the customer requirements.

Figure 71. Capacity Remaining Scoreboard Dashboard

8.1.9 Management Cluster

The Management cluster has several different components that must be monitored for continued operation of the cloud. A sample Management Dashboard is shown in Figure 72. The example monitors the following cloud management components:

- SQL Server (vCenter/vCloud Director databases).
- vCenter Server.
- vCloud Director cell.
- vCenter Operations Manager.
- Chargeback.
- vCloud Networking and Security Manager.

Figure 72. Management Cluster Dashboard

The first widget is used to monitor the SQL database for database size and transactions metrics. The Hyperic agent is used to collect these metrics and display them in a scoreboard widget.

The following metrics are used for this widget:

- Metric used: Microsoft SQL Server 2008 (database `SQL01-Server MSSQLSERVER`).
- Databases: VCD and vCenter (the SQL databases).

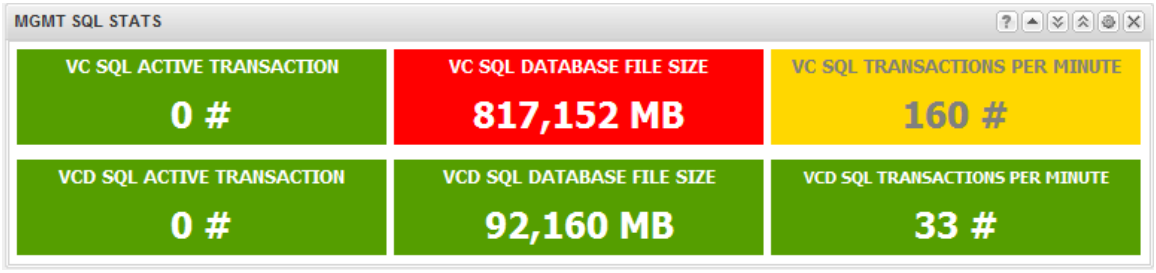
-

Metric	Unit	Green Range	Yellow Range	Orange Range	Red Range
vCenter SQL Active Transactions	#	100	100–200	200–300	300
vCenter SQL Database File Size	MB	60000-70000	70000–80000	80000–90000	900000
vCenter SQL Transactions per Minute	#	100	100–200	200–300	300
VCD SQL Active Transactions	#	100	100–200	200–300	300
VCD SQL Database File Size	MB	10000	10000–20000	20000–30000	30000
VCD SQL Transactions per Minute	#	100	100–200	200–300	300

Note: These threshold/ranges are only examples. These values should be based on the cluster design threshold values and the customer requirements.

The completed widget appears as follows:

Figure 73. Completed Widget



The following widget monitors the SQL Server free space on each logical operating system drive. This example uses the following drive letters:

- C:\ Operating system driv.
- D:\ SQL database files.
- E:\ SQL log files.
- F:\ SQL backup.

This example monitors free space available on each of these drives and configures alerts based on thresholds. The metric used is part for the resource kind *Virtual Machine* and data is collected using VMware Tools. This example uses the Scoreboard widget as follows.

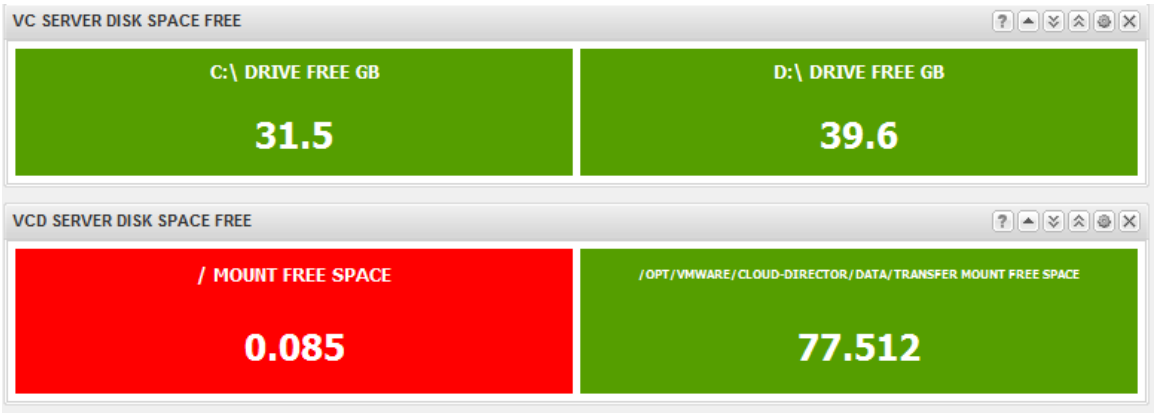
Metric	Unit	Green Range	Yellow Range	Orange Range	Red Range
Guest File System C:\ Guest File System Free	GB	15	10–15	8–10	8
Guest File System D:\ Guest File System Free	GB	15	10–15	8–10	8
Guest File System E:\ Guest File System Free	GB	15	10–15	8–10	8
Guest File System F:\ Guest File System Free	GB	15	10–15	8–10	8

Note: These threshold/ranges are only examples. These values should be based on the cluster design threshold values and the customer requirements.

The completed widget appears as follows:

Figure 74. Completed Widget xxxxx

The same metric can be used to create a widget that monitors the vCenter and vCloud Director cells for free space. Monitor the transfer folder free space of the vCloud Director cells. Using a Scoreboard widget you can create a widget as shown in the following illustrations. Note that the transfer folder mount point is monitored. The default mount point is `/opt/vmware/cloud-director/data/transfer` in the Linux operating system (vCloud Director cell).



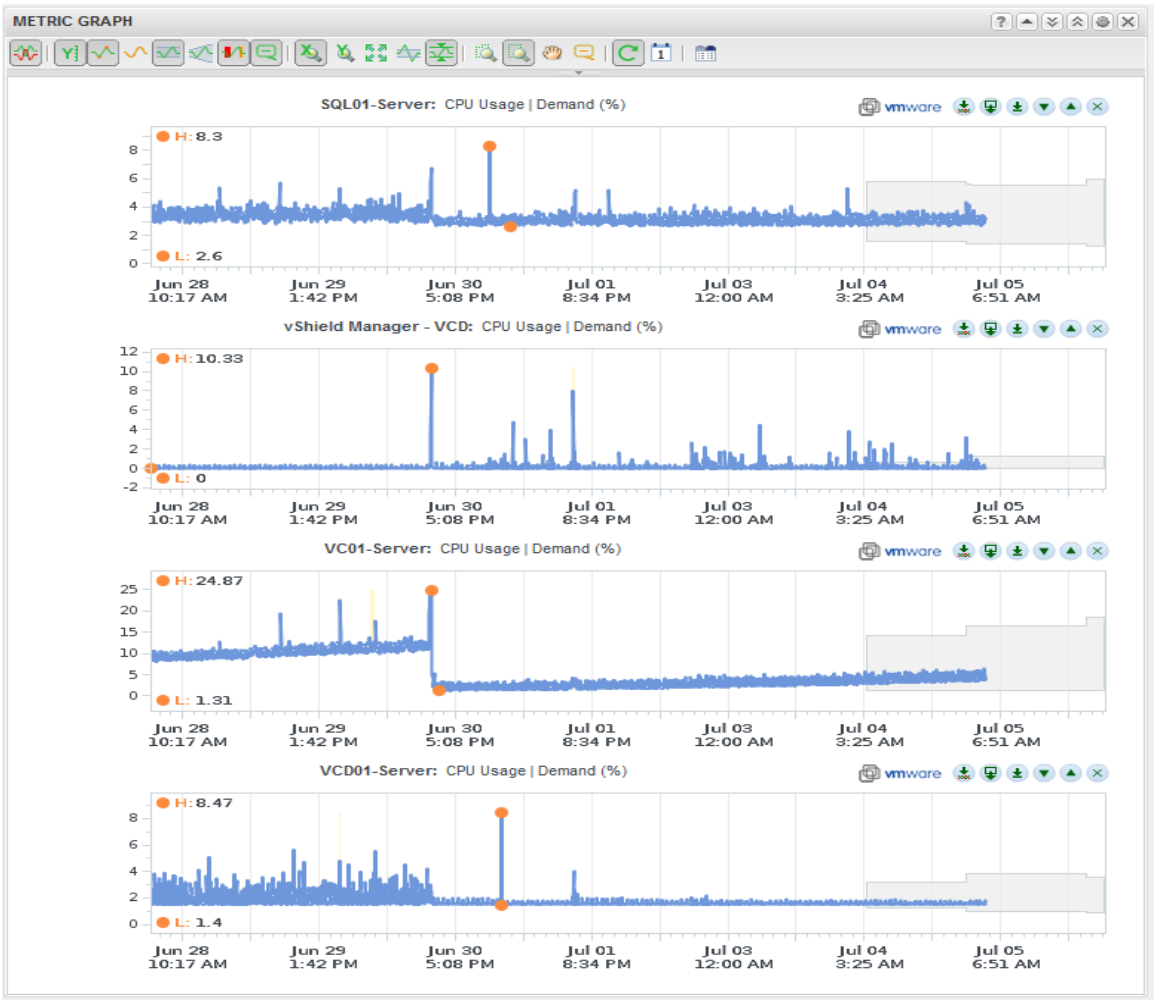
You can use a Metric Graph widget to monitor for historical CPU demand on the following infrastructure objects:

- SQL Server
- vCloud Networking and Security Manager
- vCloud Director cells
- vCenter Server

The following metrics are used to create the widget.

Resource	Metric
SQL Server	CPU Usage | Demand (%)
vCloud Networking and Security Manager	CPU Usage | Demand (%)
vCenter Server	CPU Usage | Demand (%)
vCloud Director Cell	CPU Usage | Demand (%)

The completed widget displays as follows:



8.2 AMQP Messages

Deployment Models: private.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1, RabbitMQ 2.8.4.

8.2.1 Background

vCloud Director (VCD) uses a message bus architecture for communicating VCD events with third-party systems or services. Events fall into two classes, non-blocking and blocking. In the case of non-blocking events, VCD publishes every action taken as a message to the configured exchange queue. Non-blocking messages do not halt the generating task and the task continues on regardless of what processing or acknowledgement is done on the event message.

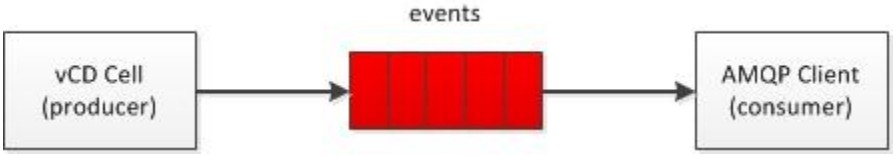
Non-blocking and blocking event messages are not multitenant in nature and are not exposed to the tenant layer of the cloud. They are inherently intended for use by the cloud provider and not the tenant.

8.2.2 Example

The Enterprise cloud administrator would like to collect all events generated by vCloud Director (VCD) to store them for searching later to support troubleshooting or audit activities within the environment.

A non-blocking messages or notifications configuration consists of the following components:

- vCloud Director cell (message producer)
- AMQP 0.9 or later message bus (message queue)
- AMQP client (message consumer)



The producer (VCD) sends messages to the *events* queue. The client (consumer) receives messages from the queue.

There are several ways of defining the queue type based on the use case for the messages being delivered. Message queues can be defined as direct, topic, fan out, system, or header.

For information on the exchange types and how to configure binding/routing of messages refer to the *RabbitMQ Tutorials* (http://www.rabbitmq.com/getstarted.html).

8.2.2.1. AMQP Server Exchange Configuration

For detailed AMQP configuration with Rabbit MQ, refer to *Downloading and Installing RabbitMQ* (http://www.rabbitmq.com/download.html).

vCloud Director (VCD) requires an AMQP 0.9 or later compatible message bus. This example uses Rabbit MQ as the message bus. Rabbit MQ provides a polyglot messaging infrastructure, with clients for all of the latest development languages as well as generic APIs like HTTP.

Table 21. RabbitMQ Server Exchange Configuration

Configuration Item	Value
AMQP Host	192.168.1.100
AMQP Port	5672
Exchange	vcdExchange
Exchange Type	topic
Durability	durable
Auto Delete	no
Internal	no
vHost	/
Prefix	vcd

8.2.2.2. AMQP Server Queue Configuration

This example creates a queue bound to this exchange for the messages to be routed into.

Table 22. RabbitMQ Server Queue Configuration

Configuration Item	Value
Queue	notificationQueue
Durability	Durable
Auto Delete	No
Exchange	vcdExchange
Routing Key	#

8.2.2.3. Exchange Routing

The AMQP broker uses routing as a way to filter vCloud Director notification messages and send them to the appropriate queue for multiple consumers. For example, a public cloud provider can filter messages based on organization and send each customers notifications to a separate queue for isolation of logging information. The vCloud Director routing key syntax is as follows:

```
<operationSuccess>.<entityUUID>.<orgUUID>.<userUUID>.<subType1>.<subType2>….<subtypeN>
.taskName
```

For example, to route only *VM create* messages to a queue, the routing key would be:

```
true.#.com.vmware.vcloud.event.vm.create
```

vCloud Director sets sane routing keys in the messages that are generated. This example uses the # (hash) routing key because this is a wildcard match on one or more segments of a routing key. This effectively routes all messages generated by vCloud Director of type vm.create to a notificationQueue. If you are interested in specific messages being routed to the appropriate queue, a non-wildcard or selective wildcard (*) routing key can be used.

Blocking tasks messages have similar identifier with the object being the blocking task. The blocking task references the following:

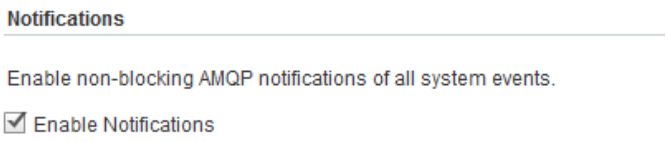
- Its parent task – The suspended task referencing the object and the task parameters attributes it was set with in the original request.
- TaskOwner – The object on which the task operates.
- The actions that can be taken on this blocking task (resume, abort, fail, updateProgress).

Receiving and acting upon on the blockings task is accomplished with the vCloud director API callbacks. System admin privileges are required to perform these operations.

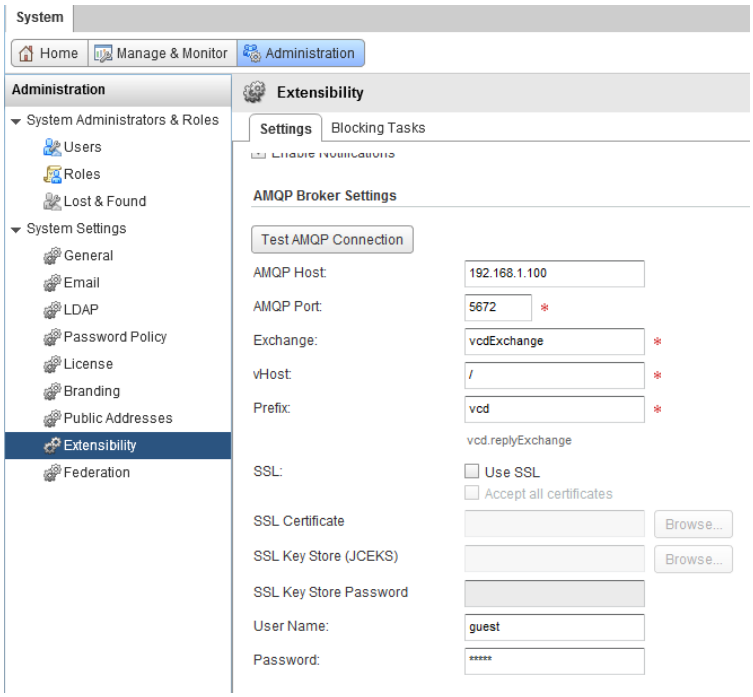
8.2.2.4. vCloud Director Configuration

To configure vCloud Director

1. Under the **System** > **Administration** > **Extensibility** section of the admin user interface, enter the appropriate configuration details for your AMQP message bus.
2. Select **Enable Notifications**.



3. vCloud Director (VCD) must be configured with the message bus settings, so that it knows where to send event messages. This is a system task and must be done by a cloud provider administrator user.



4. Click **Apply**.

VCD begins to send non-blocking notification events to the configured exchange. Regardless of whether the notification message is delivered successfully or acknowledged, the task within VCD that generated the message continues uninterrupted.

8.2.2.5. AMQP Client Configuration

Because AMQP messaging is a polyglot messaging system, the client configuration is implementation specific. Several options exist for consuming the AMQP Exchange for non-blocking messages including using the AMQP plug-in for vCenter Orchestrator or the vCloud Director API. When consuming messages, if your queue is not configured for auto-delete, you must acknowledge the messages sent to the queue when consuming them. The payload of the message is an XML document as a UTF-8 encoded string.

The following simple (Java) example loops over a queue, waiting on the delivery of the next message. ackMessage is a Boolean variable setting whether to acknowledge the message when it is retrieved from the queue.

```
@SuppressWarnings("unchecked")
public void retrieveMessages() throws IOException, InterruptedException {
    Channel channel = connection.createChannel();
    QueueingConsumer consumer = new QueueingConsumer(channel);
    channel.basicConsume(amqpConnection.getQueue(), ackMessage, consumer);

    while (true) {
        QueueingConsumer.Delivery delivery = consumer.nextDelivery();
        Map headers = delivery.getProperties().getHeaders();
        String payload = new String(delivery.getBody(), "UTF8");
```

This example performs the basic steps to acknowledge and process the AMQP messages to obtain the Message Headers and Message Body from a single Message Queue and set the Message Body to a String for subsequent processing. Additional logic could be used to take action based on the Message Header or Message Body.

8.3 AMQP Blocking Tasks

Deployment Models: private, public, hybrid.

Example Components: vCloud Director 5.1, vCloud Networking and Security Edge 5.1, vSphere 5.1, RabbitMQ 2.8.4.

8.3.1 Background

vCloud Director (VCD) utilizes a message bus architecture for communicating VCD events with third-party systems or services. Events fall into two classes, non-blocking and blocking. In the case of blocking events, VCD publishes messages to the configured exchange queue for the tasks that are configured as blocking tasks. Blocking tasks halt the generating task and the task waits for acknowledgement that it is allowed to continue.

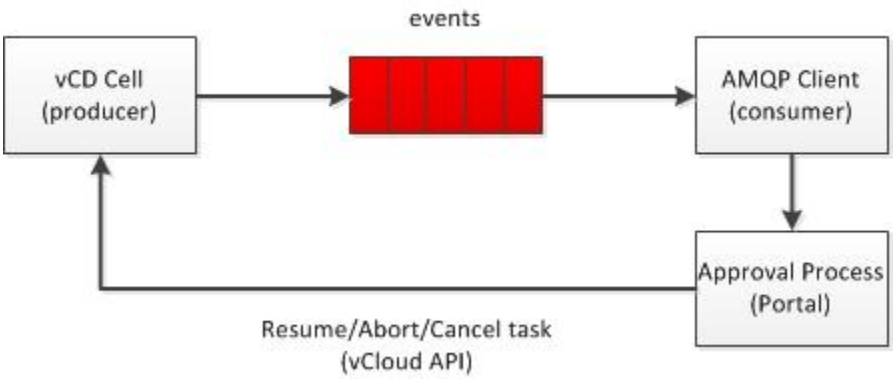
8.3.2 Example

In this example, the private cloud provider would like to configure approvals for the Instantiation of vApps by cloud consumers in their private clouds. When a user attempts to Instantiate a vApp from a template, vCloud Director sends a notification of the request to the appropriate AMQP exchange, and waits on acknowledgement and approval of the request before continuing.

The blocking task example configuration shown in the following figure consists of the following components:

- vCloud Director 5.1 cell (message producer).
- AMQP 0.9 or later message bus (message queue).
- AMQP client (message consumer).
- REST client or vCloud SDK.

Figure 75. AMQP Blocking Task Architecture



The producer (VCD) sends messages to the *events* queue. The client (consumer) receives messages from the queue. The client triggers the approval process. The result of the approval step (resume, abort, cancel) is submitted to VCD for completion of the task within VCD. At this point the task completes its processing or is terminated with the appropriate state.

There are several ways of defining the queue type based on the use case for the messages being delivered. Message queues can be defined as direct, topic, fan out, system or header.

For information on the exchange types and how to configure binding/routing of messages refer to the *RabbitMQ Tutorials* (http://www.rabbitmq.com/getstarted.html).

8.3.2.1. Exchange Configuration

vCloud Director (VCD) requires an AMQP 0.9 or later compatible message bus. This example uses Rabbit MQ as the message bus. Rabbit MQ provides a polyglot messaging infrastructure, with clients for all of the latest development languages as well as generic APIs like HTTP.

For detailed AMQP configuration with Rabbit MQ, refer to *Downloading and Installing RabbitMQ* (http://www.rabbitmq.com/download.html).

Table 23. RabbitMQ Server Exchange Configuration

Configuration Item	Value
AMQP Host	192.168.1.100
AMQP Port	5672
Exchange	vcdExchange
Exchange Type	topic
Durability	durable
Auto Delete	no
Internal	no
vHost	/

Prefix	vcd

8.3.2.2. RabbitMQ Queue Configuration

This example creates a queue bound to this exchange for the messages to be routed into.

Table 24. AMQP Queue Configuration

Configuration Item	Value
Queue	notificationQueue
Durability	Durable
Auto Delete	No
Exchange	vcdExchange
Routing Key	#

8.3.2.3. Exchange Routing

The AMQP broker uses routing as a way to filter vCloud Director notification messages and send them to the appropriate queue for multiple consumers. The vCloud Director routing key syntax is as follows:

```
<operationSuccess>.<entityUUID>.<orgUUID>.<userUUID>.<subType1>.<subType2>….<subtypeN>
.taskName
```

For example, to route only *VM create* messages to a queue, the routing key would be as follows

```
true.#.com.vmware.vcloud.event.vm.create
```

vCloud Director sets sane routing keys in the messages that are generated. This example using the # (hash) routing key because this is a wildcard match on one or more segments of a routing key. This effectively routes all messages generated by vCloud Director of type vm.create to the notificationQueue. If you are interested in specific messages being routed to the appropriate queue, a non-wildcard or selective wildcard (*) routing key can be used.

Blocking tasks messages have similar identifier with the object being the blocking task. The blocking task references the following:

- Its parent task – The suspended task referencing the object and the task parameters attributes it was set with in the original request.
- TaskOwner – The object on which the task operates.
- The actions that can be taken on this blocking task (resume, abort, fail, updateProgress).

Receiving and acting upon on the blockings task is accomplished with the vCloud director API callbacks.System admin privileges are required to perform these operations.

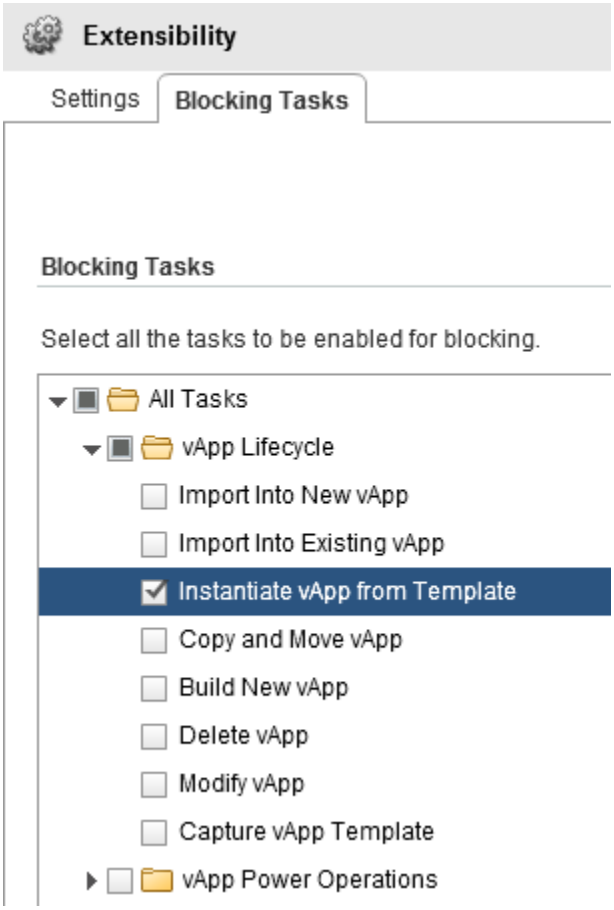
8.3.2.4. vCloud Director Configuration

Under the System->Administration->Extensibility section of the admin user interface, enter the appropriate configuration details for your AMQP message bus.

8.3.2.5. Enable Blocking Tasks

Select the Blocking Tasks Tab, and from the folder tree, select the blocking tasks you wish to enable. In this example, select the **Instantiate vApp from Template** option and click **Apply**, as shown in the following figure.

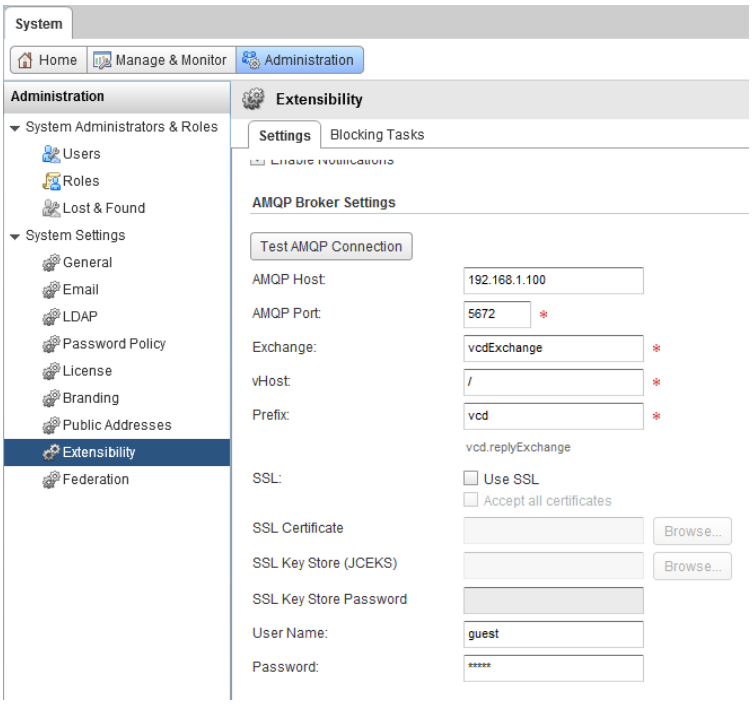
Figure 76. Enable a Task for Blocking in vCloud Director



8.3.2.6. Message Bus Configuration

vCloud Director (VCD) must be configured with the message bus settings, so it knows where to send event messages. This is a system task and must be done by a cloud system administrator. After completing these steps as shown in Figure 77, click **Apply**.

Figure 77. vCloud Director AMQP Configuration



8.3.2.7. Blocking Message

The blocking message contains a reference to the task submitted so that the approver process can locate the task to interact with it.

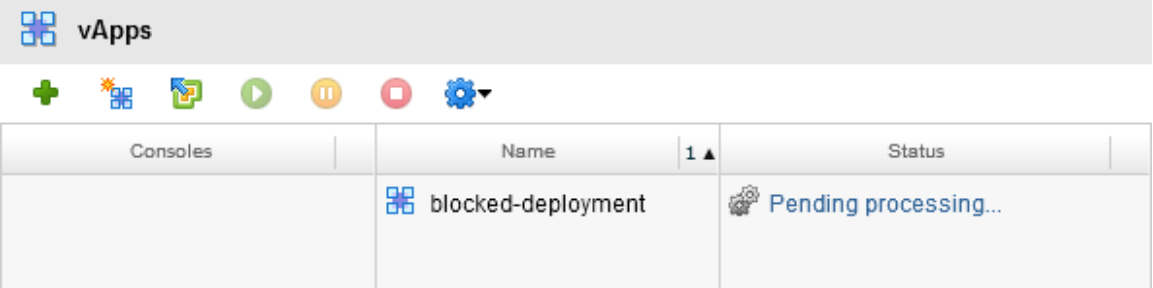
```
    <vmext:Link rel="entityResolver" href="https://192.168.1.44/api/entity/"/>
    <vmext:EntityLink rel="entity" type="vcloud:blockingTask"
name="vdcInstantiateVapp" id="urn:vcloud:blockingTask:9f4b1051-7c44-40e7-b0da-
49e611b551be"/>
    <vmext:EntityLink rel="down" type="vcloud:user" name="system"
id="urn:vcloud:user:8b209f7f-052f-41e0-bba3-063aab1d7b04"/>
    <vmext:EntityLink rel="up" type="vcloud:org" name="nuvemo"
id="urn:vcloud:org:1f6de3ed-aad9-418e-95ef-ac93bcf2b774"/>
    <vmext:EntityLink rel="task" type="vcloud:task" name="vdcInstantiateVapp"
id="urn:vcloud:task:5f1a2884-fac0-4b3d-ae50-8cd8bd7090e7"/>
    <vmext:EntityLink rel="task:owner" type="vcloud:vapp"
id="urn:vcloud:vapp:f12509a8-71d1-4484-8062-b444c7aae6e2"/>
```

8.3.2.8. Approval Process Implementation

The approval process consists of sending a blocking message, performing the approval action such as sending an email or creating a webform, and then based on the result of that action resuming or failing the task that was blocked.

8.3.2.9. Deploying a vApp

When the vApp is initially deployed, because we have configured a blocking task for **Instantiate vApp from Template**, upon deployment the vApp enters into a **Pending processing** state.



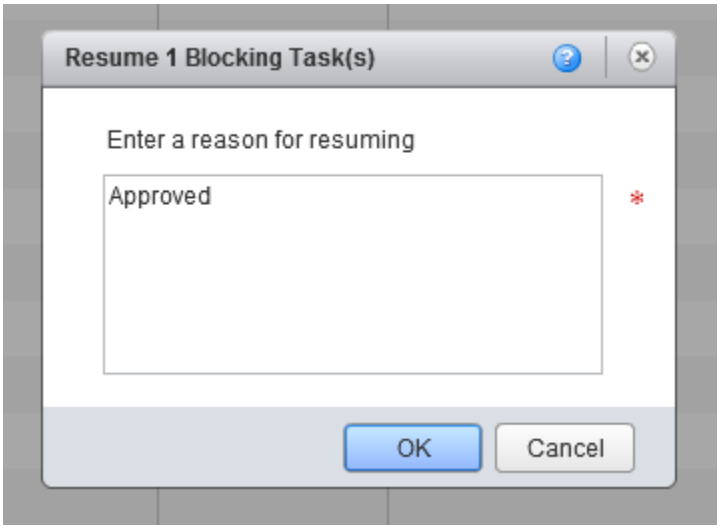
8.3.2.10. Resuming a Blocked Task

Blocked tasks can be resumed either via the vCloud Director GUI as a cloud provider administrator, or programmatically using the vCloud API.

8.3.2.11. Administrator Resume

In the **System** > **Blocking Tasks** window, the cloud administrator has a list of blocking tasks and their statuses. Right-click the task, or highlight the task and clicking the options icon, to give the administrator the options to resume, abort, or fail the blocked task.

The window prompts to enter a reason for the status change to the blocked task.



After being resumed, the vApp continues deployment and will complete unless any additional blocked tasks relevant to other deployment steps halt the process.

8.3.2.12. vCloud API Resume

To use the API

1. Find the blocked task from the message with the entity resolver using the following command:

   ```
   GET https://vcd51-01.corp.nuvemo.com/api/entity/urn:vcloud:blockingTask:9f4b1051-7c44-40e7-b0da-49e611b551be
   ```

   The data returned is the blocking task entity:

   ```
   <Entity xmlns="http://www.vmware.com/vcloud/v1.5" name="urn:vcloud:blockingTask:9f4b1051-7c44-40e7-b0da-49e611b551be" id="urn:vcloud:blockingTask:9f4b1051-7c44-40e7-b0da-49e611b551be" type="application/vnd.vmware.vcloud.entity+xml" href="https://vcd51-01.corp.nuvemo.com/api/entity/urn:vcloud:blockingTask:9f4b1051-7c44-40e7-b0da-49e611b551be" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.vmware.com/vcloud/v1.5 http://192.168.1.44/api/v1.5/schema/master.xsd">
       <Link rel="alternate" type="application/vnd.vmware.admin.blockingTask+xml" href="https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-7c44-40e7-b0da-49e611b551be"/>
   </Entity>
   ```

2. Get the blocking task from the resolved entity

   ```
   GET https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-7c44-40e7-b0da-49e611b551be
   ```

   This returns the blocking task and methods that can be performed against it:

   ```
   HTTP/1.1 200 OK
   Date: Tue, 03 Jul 2012 16:30:33 GMT
   Date: Tue, 03 Jul 2012 16:30:33 GMT
   Content-Type: application/vnd.vmware.admin.blockingtask+xml;version=1.5
   Content-Length: 2428


   <?xml version="1.0" encoding="UTF-8"?>
   <vmext:BlockingTask xmlns:vmext="http://www.vmware.com/vcloud/extension/v1.5" xmlns:vcloud="http://www.vmware.com/vcloud/v1.5" status="active" timeoutDate="2012-07-08T09:06:33.757-07:00" timeoutAction="abort" createdTime="2012-07-03T09:06:33.757-07:00" name="vdcInstantiateVapp" id="urn:vcloud:blockingTask:9f4b1051-7c44-40e7-b0da-49e611b551be" type="application/vnd.vmware.admin.blockingTask+xml" href="https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-7c44-40e7-b0da-
   ```

```
49e611b551be" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.vmware.com/vcloud/extension/v1.5
http://192.168.1.44/api/v1.5/schema/vmwextensions.xsd
http://www.vmware.com/vcloud/v1.5 http://192.168.1.44/api/v1.5/schema/master.xsd">

    <vcloud:Link rel="resume"
type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
href="https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-
7c44-40e7-b0da-49e611b551be/action/resume"/>

    <vcloud:Link rel="abort"
type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
href="https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-
7c44-40e7-b0da-49e611b551be/action/abort"/>

    <vcloud:Link rel="fail"
type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
href="https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-
7c44-40e7-b0da-49e611b551be/action/fail"/>

    <vcloud:Link rel="updateProgress"
type="application/vnd.vmware.admin.blockingTaskUpdateProgressOperationParams+xml"
href="https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-
7c44-40e7-b0da-49e611b551be/action/updateProgress"/>

    <vcloud:Link rel="up" type="application/vnd.vmware.vcloud.task+xml"
href="https://vcd51-01.corp.nuvemo.com/api/task/5f1a2884-fac0-4b3d-ae50-
8cd8bd7090e7"/>

    <vcloud:Organization type="application/vnd.vmware.admin.organization+xml"
name="nuvemo" href="https://vcd51-01.corp.nuvemo.com/api/admin/org/1f6de3ed-aad9-
418e-95ef-ac93bcf2b774"/>

    <vcloud:User type="application/vnd.vmware.admin.user+xml" name="system"
href="https://vcd51-01.corp.nuvemo.com/api/admin/user/8b209f7f-052f-41e0-bba3-
063aab1d7b04"/>

    <vcloud:TaskOwner type="application/vnd.vmware.vcloud.vApp+xml" name=""
href="https://vcd51-01.corp.nuvemo.com/api/vApp/vapp-f12509a8-71d1-4484-8062-
b444c7aae6e2"/>

</vmext:BlockingTask>
```

3. Resume the blocked task

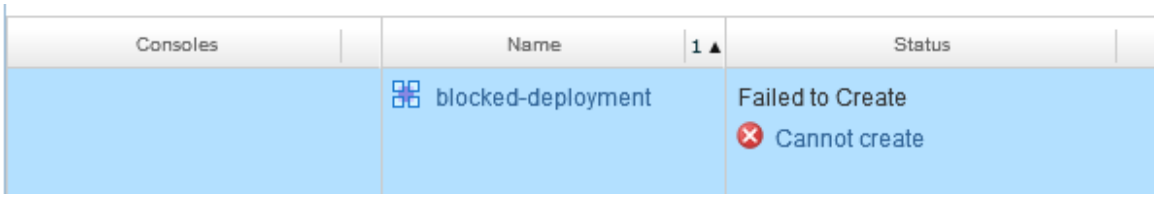
```
POST https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-
7c44-40e7-b0da-49e611b551be/action/resume

Content-Type: application/vnd.vmware.admin.blockingTaskOperationParams+xml
```

Pass in this content as the post to the resume request:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<BlockingTaskOperationParams
  xmlns=http://www.vmware.com/vcloud/extension/v1.5
  <Message>Approved task. </Message>
</BlockingTaskOperationParams>
```

8.3.2.13. Failed/Aborted Task

The end user (tenant) is notified through the vCloud Director GUI that the deployment has failed or has been aborted. Click into the details of the failure message to find the provider entered reason for failing, which is presented in the details.



Failing or aborting tasks is carried out in the same manner via the vCloud API as a resume task. Looking at the XML returned when performing a GET on the blocking task, methods are returned for Resume, Abort, and Fail, as follows:

```
    <vcloud:Link rel="abort"
type="application/vnd.vmware.admin.blockingTaskOperationParams+xml"
href="https://vcd51-01.corp.nuvemo.com/api/admin/extension/blockingTask/9f4b1051-7c44-
40e7-b0da-49e611b551be/action/abort"/>
```

You can fail or abort a task as part of a provisioning process that requires an external approval to complete. If you have a provisioning portal or workflow executing the task of deploying and approving vApps, when an approval is declined, the portal fails the blocked task with an appropriate message to the user. If a user submits a deploy request for a vApp, and then decides to cancel the deployment prior to the approval, the portal can issue an abort on the blocking task with the appropriate message. An abort and fail both result in a termination of the vApp deployment, however contextually they are different.