



VMware vCloud[®] Architecture Toolkit

Hybrid VMware vCloud Use Case

Version 2.0.1

October 2011

© 2011 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. This product is covered by one or more patents listed at <http://www.vmware.com/download/patents.html>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

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

Contents

1. Overview	5
1.1 Business Requirements	6
1.2 Use Case	6
2. vSphere and Public vCloud Hybrid Scenario	7
2.1 vSphere Private Resources	8
2.2 vCloud Public Resources	8
2.3 Configuring Secure Connectivity to the Public vCloud	9
2.4 Triggering Techniques for Cloudburst into the Public vCloud	10
2.5 Scaling of the Front-End Logic	11
2.6 Scaling the Front-End into the Public vCloud Through VMware vCloud Connector ...	12
2.7 Load Balancer Configuration	13
2.8 Decommissioning Public Cloud Resources	15
3. Private vCloud and Public vCloud Hybrid Scenario	16
3.1 vCloud Private Resources	17
3.2 vCloud Public Resources	17
3.3 Configuring Secure Connectivity to the Public vCloud	18
3.4 Triggering Techniques for Cloudburst into the Public vCloud	20
3.5 Scaling of the Front-End Logic	20
3.6 Scaling the Front-End into the Public vCloud Through APIs	21
3.7 Load Balancer Configuration	22
3.8 Decommissioning Public vCloud Resources	24

List of Figures

Figure 1. High-Level Design (vSphere and Public vCloud) 7

Figure 2. vcloud.vmware.com Sample Page 9

Figure 3. VPN Configuration Details (vSphere and Public vCloud) 9

Figure 4. Scheduled Burst Period 11

Figure 5. vCloud Connector Architecture 12

Figure 6. Typical Internet Service Deployment at NewCo 13

Figure 7. Load Balancing During the Burst (from a Local vSphere Infrastructure) 14

Figure 8. High-Level Design (Private vCloud and Public vCloud) 16

Figure 9. vcloud.vmware.com Sample Page 18

Figure 10. VPN Configuration Details (Private vCloud and Public vCloud) 19

Figure 11. Scheduled Burst Period 20

Figure 12. Typical Private vCloud Service Deployment at NewCo 22

Figure 13. Load Balancing During the Burst (from a Private vCloud) 23

List of Tables

Table 1. Solution Components (vSphere and Public vCloud) 8

Table 2. Infrastructure IP Addresses Details (vSphere and Public vCloud) 10

Table 3. Online Service IP Address Details (vSphere and Public vCloud) 15

Table 4. High-Level Design (Private vCloud and Public vCloud) 17

Table 5. Infrastructure IP Addresses Details (vSphere and Public vCloud) 19

Table 6. Online Service IP Address Details (Private vCloud and Public vCloud) 23

1. Overview

There are many potential use cases for a hybrid vCloud, but this hybrid VMware vCloud® use case example focuses on scenarios where a customer with a Web presence needs to extend (*cloudburst*) their front-end server farm due to anticipated spikes in incoming traffic. The example outlines what the customer can implement in terms of federating the local private vCloud with a set of resources that are available in the public vCloud.

Although this is not the most common and easiest use case for most customers (at least initially), it serves to describe the potential of the hybrid concepts.

The use case example covers two slightly different scenarios:

- The first scenario assumes a VMware customer with an on-premise VMware vSphere® deployment. This customer wants to federate and extend the local vSphere setup using public vCloud resources. This is probably the most typical scenario given the large number of vSphere deployments in the market.

Note Though the industry generally refers to a hybrid cloud as a combination of private and public resources, this scenario includes a traditional vSphere infrastructure on-premise (as opposed to a true private cloud).

- The second scenario assumes a VMware customer with an on-premise private vCloud deployment. This customer wants to federate and extend the private vCloud setup using public vCloud resources.

Both scenarios assume a vCloud-based public vCloud as a remote, online resource, and cover the following:

- Subscribing to a public vCloud for capacity overflow.
- Configuring secure connectivity to the public vCloud.
- Triggering techniques for cloudburst into the public vCloud.
- Cloning of the front-end logic.
- Moving clones into the public vCloud.
- Reconfiguring the infrastructure to drive end-user requests through the public vCloud resources.
- Decommissioning public vCloud resources.

The two scenarios are consistent and self-contained—shared concepts are repeated in both sections.

This hybrid vCloud use case example is written more from a consumption perspective than a deployment perspective because a hybrid vCloud is really about tying private and public vCloud services together in a virtual continuum of homogenous resources. This example focuses more on connecting to and consuming resources as uniformly as possible.

This document also provides hyperlinks to private and public vCloud implementation details that are available in other publications.

1.1 Business Requirements

New Company (NewCo) is a relatively small online media company that does a large part of their business over the Web. For NewCo, it is critical to have an infrastructure that can give them a way to contain costs through operational efficiency, but which also gives them the agility and responsiveness required in the Internet-facing world. In this context, their customers are spread across the globe, service quality expectations are high, and their competitors are only a click away.

1.2 Use Case

NewCo is launching a new online service. The marketing and sales departments are not able to provide a reliable forecast of customers that will be using the service, and they have warned the IT organization that visibility in the market for this service will be subject to a high degree of variability with very high peaks that coincide with specific marketing campaigns.

This online service has peculiar technical requirements. The nature of the application is such that the solution is to be considered CPU and memory bound at the front-end layer. NewCo noted, during QA, that each Web instance can support a relatively small number of connections before processor and RAM resources are exhausted. This could slow down additional users connected to the same instance, or sometimes throw exceptions.

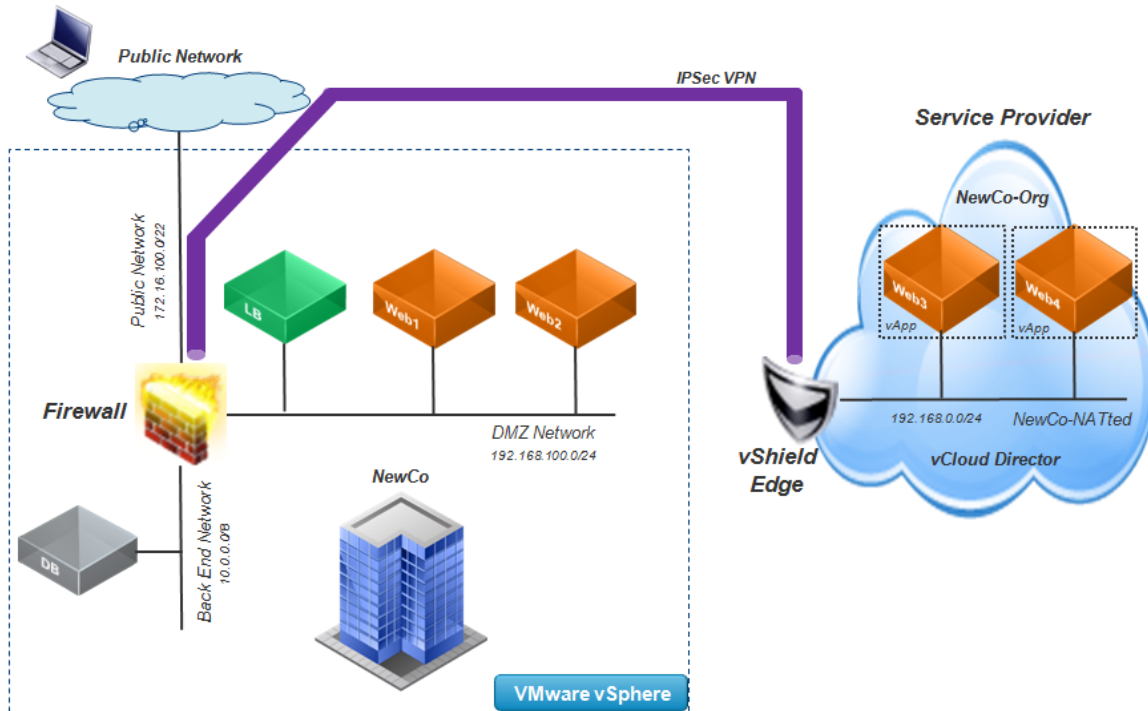
Ideally, the application should be revisited to remove these scalability issues, but NewCo has no time to do that before the new service becomes available, so they chose to add capacity to cope with this problem. NewCo does not want to size their local IT infrastructure to cope with the anticipated peaks because the peaks usually last only a few weeks and the additional infrastructure resources would be idle most of the time. Outsourcing a dedicated infrastructure to take advantage of a service provider's resource elasticity is not an option because there is a shared backend with many other online services running on-premise, and because the backend is not uniquely tied to specific new offerings, they cannot move it off-premise. Also, some of the data in the backend is sensitive, so they want to keep it inside their firewall.

For these reasons, NewCo is evaluating an innovative hybrid deployment model that enables them to gain flexibility and, at the same time, comply with their constraints.

2. vSphere and Public vCloud Hybrid Scenario

NewCo is testing a new way to manage peak demand by federating their vSphere infrastructure with a vCloud provider in their region, essentially creating a hybrid vCloud solution. By doing so, they can transparently and securely extend the infrastructure that fronts the Web traffic generated by their customers and prospects. Figure 1 shows the high-level design.

Figure 1. High-Level Design (vSphere and Public vCloud)



NewCo’s local infrastructure is built using common datacenter patterns. They have a traditional firewall that segments their VLAN-based infrastructure. These VLANs are then trunked into VMware ESX®/VMware ESXi™ hosts and exposed to virtual machines as port groups.

- One of the VLANs supports a database server that acts as the backend for the new service (in addition to other services).
- A second VLAN is used to host the front-end logic, which is comprised of a Web service as well as middleware and complex application logic. This second VLAN also hosts the load balancer that spreads the traffic across the front-end.
- The third VLAN provides connectivity to customers from the Internet.

On the service provider side, the VLANs are connecting to a virtual datacenter with a private network connected to the Internet and protected by a VMware vShield Edge™ device. This device is the second endpoint for the VPN tunnel.

Table 1. Solution Components (vSphere and Public vCloud)

Virtual Machine	Purpose
LB	Load balancer used by this online service.
Web1	Local front-end #1.
Web2	Local front-end #2.
Web3	Remote front-end #1.
Web4	Remote front-end #2.
DB	Backend database.
Firewall	NewCo infrastructure firewall.
vShield Edge	Edge appliance backing the remote organization network.
Public network	The network from which users connect and the channel through which the VPN connection is established.

2.1 vSphere Private Resources

NewCo built their local vSphere infrastructure following field-proven best practices. For more information on how to get started, visit the VMware website or contact your local VMware partner.

2.2 vCloud Public Resources

For this use case, NewCo must subscribe access to a public vCloud. Their entry point for this is vcloud.vmware.com because it provides a single access point for customers who are interested in exploring online resources that are compatible with private vSphere and vCloud deployments. Figure 2 is a sample screenshot of the portal.

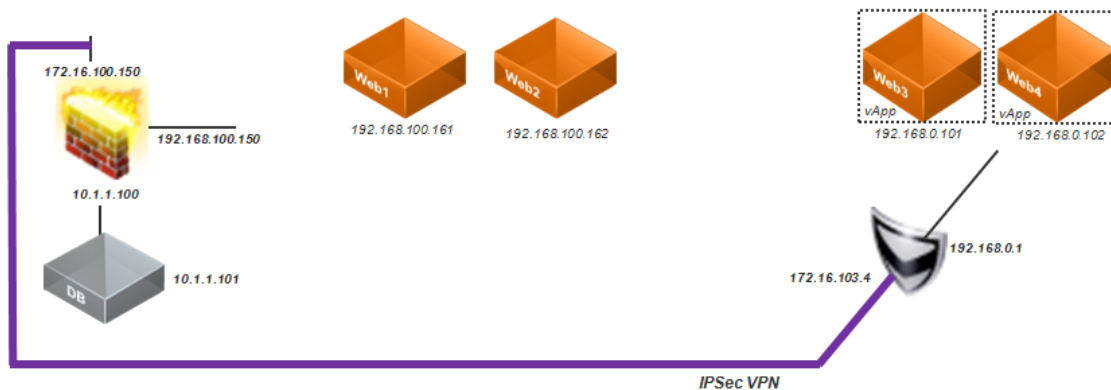
Figure 2. vcloud.vmware.com Sample Page



2.3 Configuring Secure Connectivity to the Public vCloud

The secure connectivity between the NewCo headquarters and the public vCloud is extremely important in determining the viability of the hybrid scenario. NewCo cannot accept a hybrid deployment without secure connectivity between all of the parts comprising the solution. To cover this requirement, NewCo decided to implement IPSec VPN connectivity between their local firewall with the vShield Edge device that backs the organization network where the overflow front-end is deployed. Figure 3 illustrates all of the connection points (with associated IP addresses) that NewCo is using to validate their hybrid vCloud design.

Figure 3. VPN Configuration Details (vSphere and Public vCloud)



NewCo configured their local firewall to create an IPSec VPN tunnel over the public network (in this case, the Internet) with the vShield Edge device backing their remote organization network available in the public vCloud. They determined that this was the fastest, least expensive and most flexible way to implement a secure connection with their vCloud provider. In this case, the local firewall and the remote vShield Edge allows the two subnets that host the front-end virtual machines to communicate at Layer 3 without any security enforcement other than the encrypted traffic. There are specific firewall rules that only allow these two subnets to be reached over port 443 from the public network, and other firewall rules that only allow access to the backend subnet over the database ports.

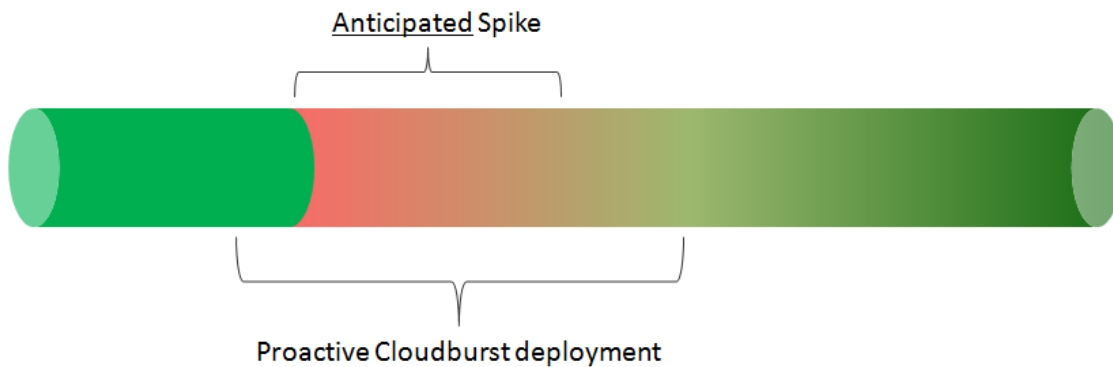
Table 2. Infrastructure IP Addresses Details (vSphere and Public vCloud)

IP Address	Purpose
172.16.100.150	Local firewall external interface (VPN endpoint)
192.168.100.150	Local firewall internal interface (web – DMZ)
10.1.1.100	Local firewall internal interface (database – backend)
172.16.103.4	Remote vShield Edge external Interface (VPN endpoint)
192.168.0.1	Remote vShield Edge internal interface (web – DMZ extension)

2.4 Triggering Techniques for Cloudburst into the Public vCloud

A cloudburst, in IT terms, is the action of scaling the capacity of an infrastructure to respond to a spike in demand.

For this hybrid scenario, NewCo does not need reaction times on the order of seconds or even minutes—they need overflow flexibility that has predictable patterns. Additionally, they determined that, when they need overflow capacity, they need that capacity only for a few days or weeks. This is because their IT needs are typically associated with marketing campaigns that are programmed up front and which last for a certain amount of days. Figure 4 illustrates this concept.

Figure 4. Scheduled Burst Period

The triggering of the burst is process-driven rather than event-driven. NewCo is not going to set up complex infrastructure and application sensors that could automatically trigger the burst event. Instead, NewCo is going to proactively deploy overflow capacity into the public vCloud a few days before the starting day of the marketing campaign so that the hybrid solution is ready to immediately accept the growing number of incoming Web requests.

Though this may not be the most advanced and efficient technique to cloudburst, it is very simple to implement, and is an effective way to meet their requirements.

2.5 Scaling of the Front-End Logic

NewCo wants to take a phased approach with their hybrid vCloud. Rather than investing up front to optimize the entire processes associated with scaling the front-end, NewCo is manually deploying additional images and moving them into the public vCloud as part of their scheduled bursting process.

The NewCo vSphere administrators are deploying additional front-end images exactly the same way as they would for local scaling requirements. This frequently happens in datacenters when the workload increases and the front-end gets scaled out. It usually involves cloning an existing application instance (or a template that is representative of that particular workload), and making the required application-specific adjustment to join the existing farm that delivers the Web service.

There is nothing in this process that is specific to vCloud—this is what IT administrators have been doing for years. What is unusual in this case is that the additional stacks being deployed will later be moved to an external infrastructure instead of being run locally (and consuming local resources).

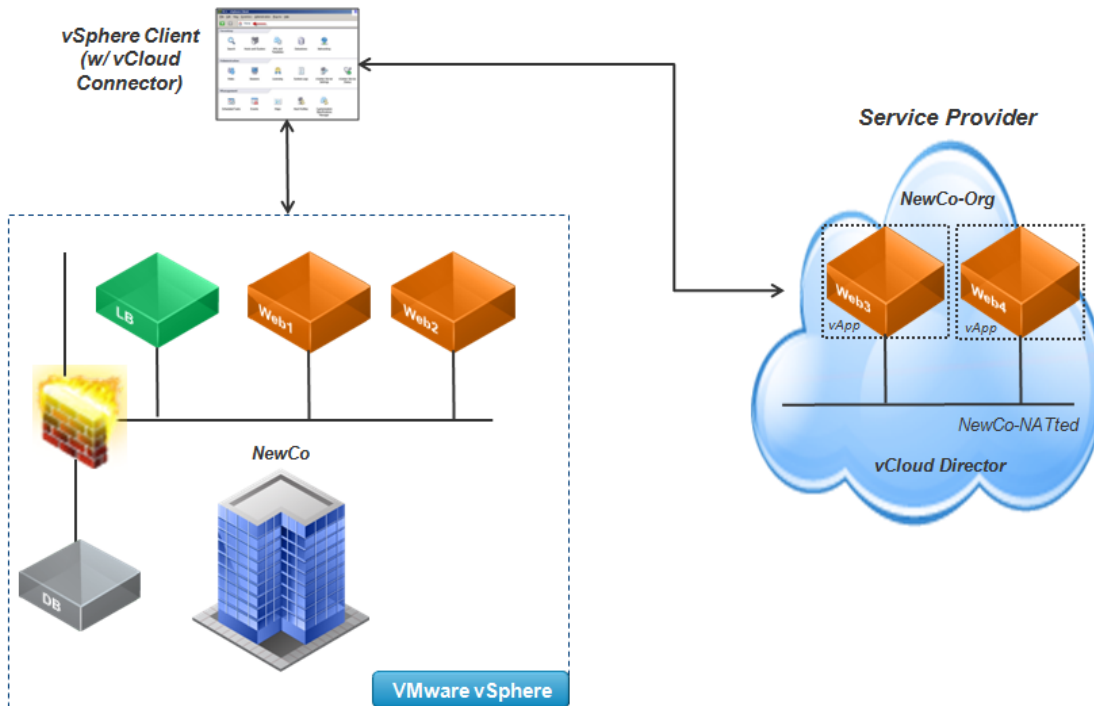
2.6 Scaling the Front-End into the Public vCloud Through VMware vCloud Connector

After NewCo instantiates additional front-end objects, there are different ways to move these virtual machines into their public vCloud organization. One method is to manually export these virtual machines from the local vSphere platform as OVF (Open Virtualization Format) files and import them into the public vCloud. However, NewCo decided to use VMware vCloud® Connector to move the objects because it allows them to export and import them more quickly and more easily, directly from the VMware vSphere® Client™. This results in deploying one vApp for each virtual machine being moved into the public vCloud. Using vCloud Connector also allows NewCo to perform basic operations on the remote virtual machines in the public vCloud. These operations can include powering them on or off, or taking control of the remote console. Being able to accomplish all of this from the vSphere interface is a bonus for NewCo.

Note that advanced networking configurations require NewCo administrators to connect to the public vCloud using the vCloud Web interfaces or the vCloud APIs.

Figure 5 shows at a high level how NewCo IT administrators are using the vSphere Client to connect to both environments.

Figure 5. vCloud Connector Architecture



NewCo is aware that the process they are using to move workloads could be significantly improved in terms of automation and bandwidth optimization. In fact, it may not be optimal to deploy all new front-end objects locally and transfer all of them into the vCloud. A better approach is to move a template into the vCloud private catalog in the NewCo-Org and deploy instances directly from the catalog.

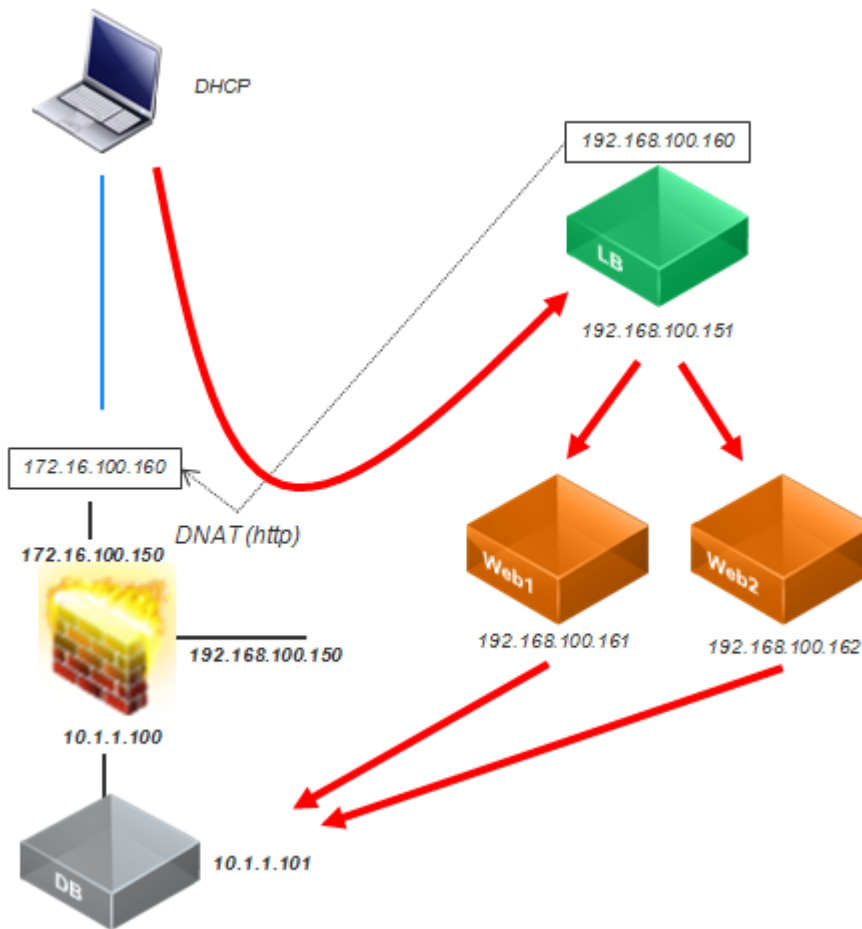
NewCo initially chose a deployment methodology that could be as similar as possible to the process they use to scale resources locally. Although this may require more time and bandwidth to set up, it significantly reduces the complexity and investments they need to make to implement the hybrid scenario. As they come to appreciate and solidify the business case for a hybrid vCloud, they will optimize the processes around it.

2.7 Load Balancer Configuration

The load balancing service is the true enabler of the hybrid solution NewCo is building. The typical local implementation is very simple and is comprised of a load balancing (local) service that spreads traffic across a number of (local) front-end servers which in turn communicate with the database when needed.

Figure 6 shows how NewCo typically deployed Internet services.

Figure 6. Typical Internet Service Deployment at NewCo

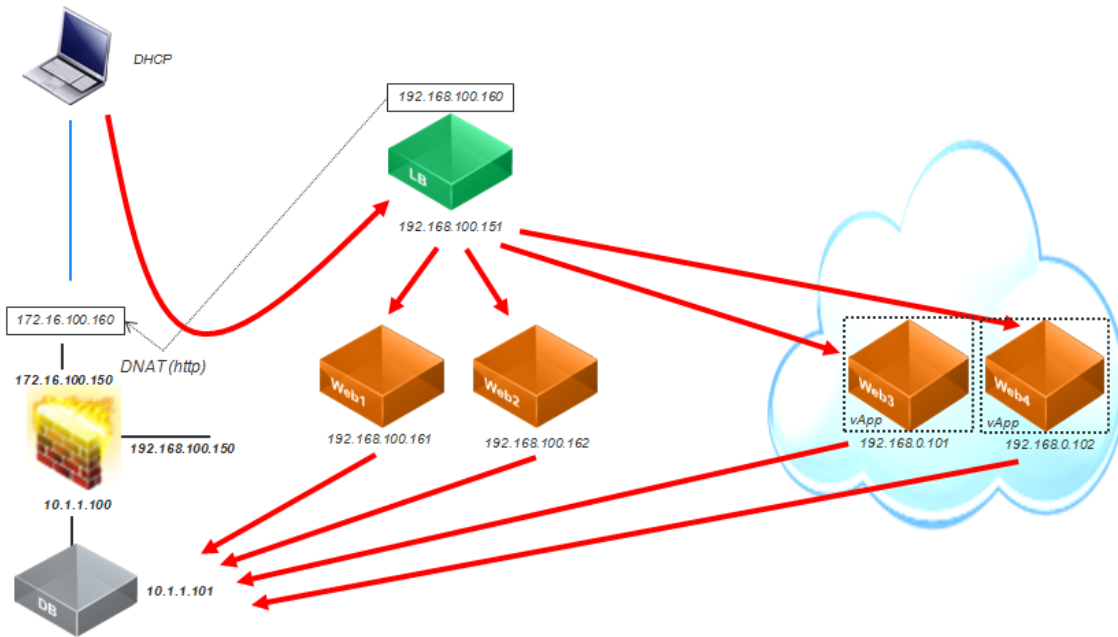


The typical NewCo setup includes a load balancer that sits in the DMZ in a one-arm configuration that balances the front-end servers. There is sufficient Internet bandwidth so that is not a constraint from a scalability perspective. Similarly, the backend database is not a constraint because they have implemented a layer of in-memory data-replication in the front-end stack that allows the application servers to touch the database only when actually committing transactions (this happens only for a small fraction of all traffic). Because of this heavy decoupling layer, and because of the scalability limitations of the application code, the front-end itself becomes a bottleneck when user traffic increases. Specifically, CPU and memory resources in the front-end layer are the limiting factors to consider when scaling horizontally.

For this reason, NewCo is not concerned about creating a global load balancing policy that optimizes bandwidth and locality. They are more concerned with having more available CPU and memory capacity to instantiate new front-end virtual machines when needed.

As a result of their analysis, NewCo decided to implement a very simple configuration that allows their load balancer to reach out to remote instances and balance them as if they were local resources. Figure 7 illustrates this approach.

Figure 7. Load Balancing During the Burst (from a Local vSphere Infrastructure)



The following table summarizes the IP addresses that were used to create this setup.

Table 3. Online Service IP Address Details (vSphere and Public vCloud)

IP Address	Purpose
172.16.100.150	NATed entry point for the online service.
192.168.100.160	Entry point for the online service (load balanced virtual IP).
192.168.100.151	Load balancer IP address.
192.168.100.161	Local front-end #1 IP.
192.168.100.162	Local front-end #2 IP.
192.168.0.101	Remote front-end #1 IP.
192.168.0.102	Remote front-end #2 IP.
10.1.1.101	Backend database.

The load balancer is configured creating a virtual server IP address that balances user requests across the two local front-end servers. During the burst the load balancer is reconfigured to add the two remote front-end servers to the list of the targets to balance. The virtual IP address is then NATed externally for the end-user to be able to connect to the Web service.

2.8 Decommissioning Public Cloud Resources

The hybrid vCloud is all about flexibility and a better cost structure. NewCo did not want to size for the peaks. They can overflow capacity into the vCloud when needed and only pay for what they use. To do so, they must decommission the resources that they instantiated for the burst period. NewCo selected the Pay-As-You-Go subscription model offered by their service provider, and decommissioning of unneeded capacity is assumed for this model.

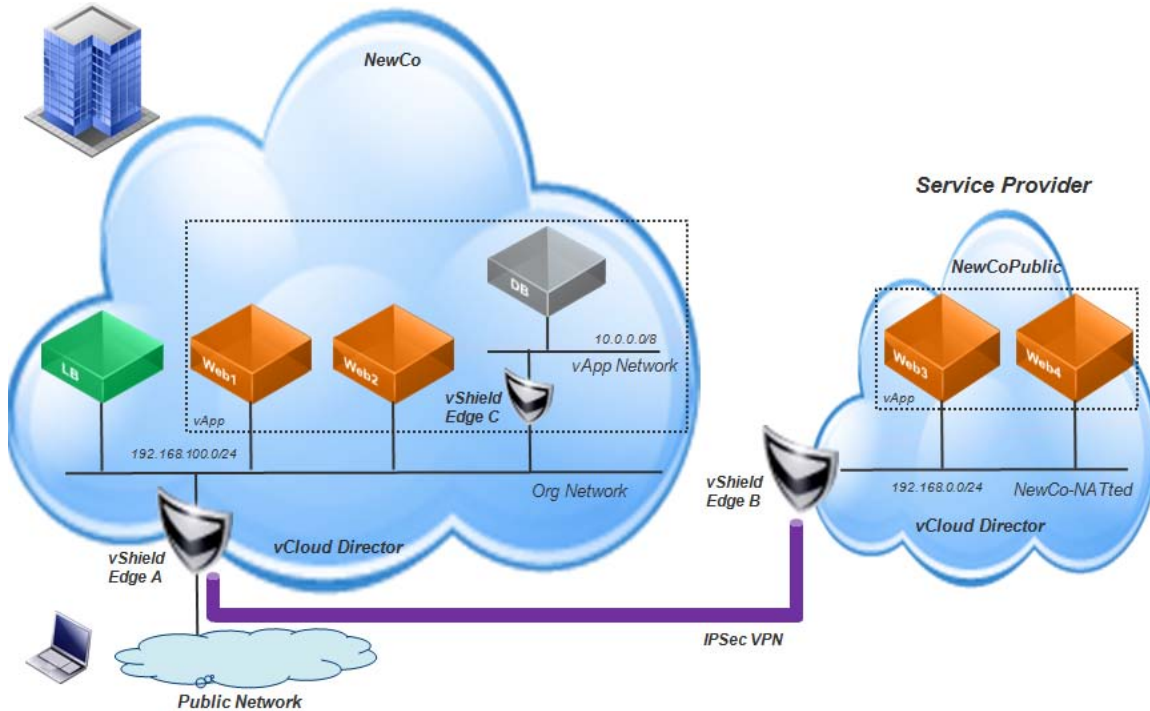
Similar to how NewCo triggers the beginning of the burst process, decommissioning is also a human-driven decision. When IT sees a slowdown in incoming end-user traffic to the online service to the point where the local front-end virtual machines can cope with the traffic, the decommissioning process can begin. This usually happens a few days after the end of the marketing campaign.

NewCo had a choice of turning off the remote instances (paying only for consumed storage) or deleting them (and not having to pay anything to the service provider). Because the front-end stack that they are using is frequently updated, and because the scheduled spikes occur only three or four times a year, NewCo decided to delete the remote instances when the spike end and copy them back over when the next burst occurs.

3. Private vCloud and Public vCloud Hybrid Scenario

NewCo is testing a new way to manage peak demand by federating their private vCloud with a vCloud provider in their region, essentially creating a hybrid vCloud solution. By doing so they can transparently and securely extend the infrastructure fronting the Web traffic generated by their customers and prospects. Figure 8 shows the high-level design.

Figure 8. High-Level Design (Private vCloud and Public vCloud)



NewCo’s local infrastructure is built using brand new vCloud paradigms. The LOB responsible for the brand new online service has a dedicated organization and a corresponding virtual datacenter that they can manage in a very flexible way. Their local organization is protected by a vShield Edge device that allows this LOB to set their own network and security configuration without having to talk to the IT department for every single infrastructure requirement.

The front-end servers are connected to the organization network and the database server is connected to a protected vApp network. This allows adding an additional level of self-service protection to the data layer. Both the front-end virtual machines and the database server are deployed in a single vApp.

The vShield Edge device backing the organization network is connected to the Internet.

On the service provider side, they are connecting to a virtual datacenter with a private network that is connected to the Internet and protected by another vShield Edge device. This device is the second endpoint for the VPN tunnel.

From this point on references to “NewCo” are specifically references to the NewCo LOB and their VMware vCloud® Director™ organization. The NewCo LOB can act as an independent entity with its own virtual datacenter.

Table 4. High-Level Design (Private vCloud and Public vCloud)

Virtual Machine	Purpose
LB	Load balancer used by this online service.
Web1	Local front-end #1.
Web2	Local front-end #2.
Web3	Remote front-end #1.
Web4	Remote front-end #2.
DB	Backend database.
vShield Edge A	Firewall for the organization in the private vCloud.
vShield Edge B	Firewall for the organization in the public vCloud.
vShield Edge C	Firewall for the database.
Public network	The network from which users connect and the channel through which the VPN connection is established.

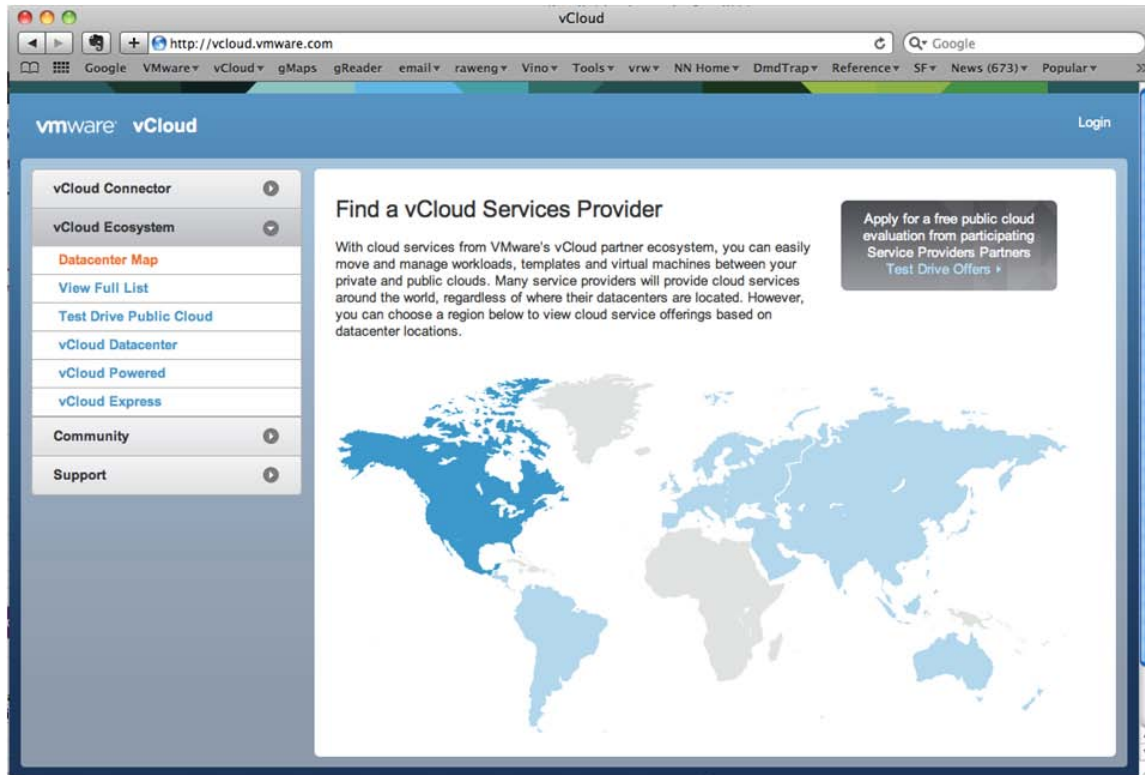
3.1 vCloud Private Resources

NewCo built their private vCloud following field-proven best practices. For more information, see the *Private VMware vCloud Implementation Example*.

3.2 vCloud Public Resources

For this use case, NewCo needs to subscribe access to a public vCloud-based vCloud. NewCo's entry point for this is ycloud.vmware.com because it provides a single access point for customers interested in exploring online vCloud resources that are compatible with private vSphere and vCloud deployments. Figure 9 is a sample screenshot of the new portal.

Figure 9. vcloud.vmware.com Sample Page

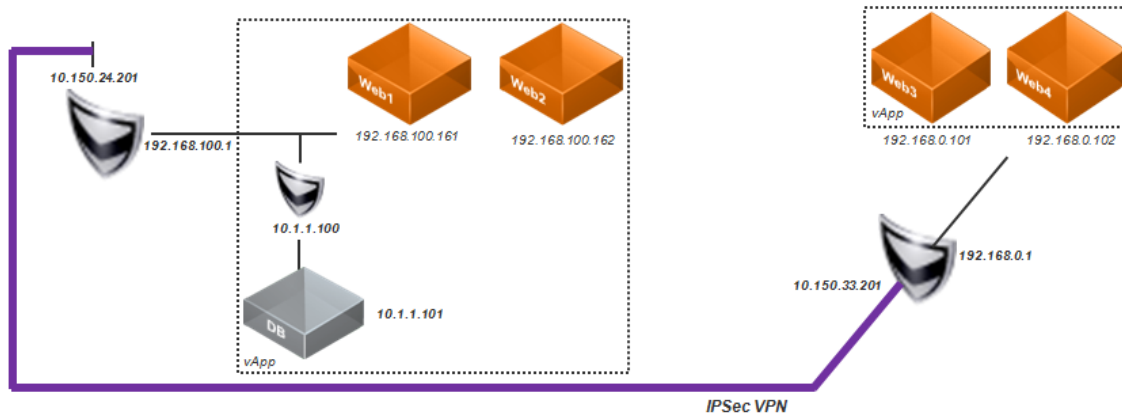


3.3 Configuring Secure Connectivity to the Public vCloud

The secure connectivity between the NewCo private vCloud and the public vCloud is extremely important in determining the viability of the hybrid scenario. NewCo cannot accept a hybrid deployment without secure connectivity between all of the parts that comprise the solution. To meet this requirement, NewCo decided to implement IPSec VPN connectivity between the VMware vShield Edge device backing the local organization of the LOB responsible for the online service and the VMware vShield Edge device backing the organization network where the overflow front-end is deployed.

Figure 10 illustrates all of the connection points (with associated IP addresses) that NewCo is using to validate their hybrid vCloud design.

Figure 10. VPN Configuration Details (Private vCloud and Public vCloud)



The NewCo LOB that is responsible for this online service has configured both their local and remote vShield Edge devices to create an IPSec VPN tunnel over the public network (in this case, the Internet) between the virtual datacenters in the private and public vClouds. They determined that this was the fastest, least expensive, and most flexible way to implement a secure connection with their vCloud provider. In this case, the local and remote vShield Edge appliances allow the two subnets that will host the front-end virtual machines to communicate at Layer 3 without any security enforcement other than the encrypted traffic. There are specific firewall rules that only allow these two subnets to be reached over port 443 from the public network, and other firewall rules (applied to the vShield Edge backing the vApp network where the database virtual machine is connected) that only allow reaching the backend subnet over the database ports.

Table 5. Infrastructure IP Addresses Details (vSphere and Public vCloud)

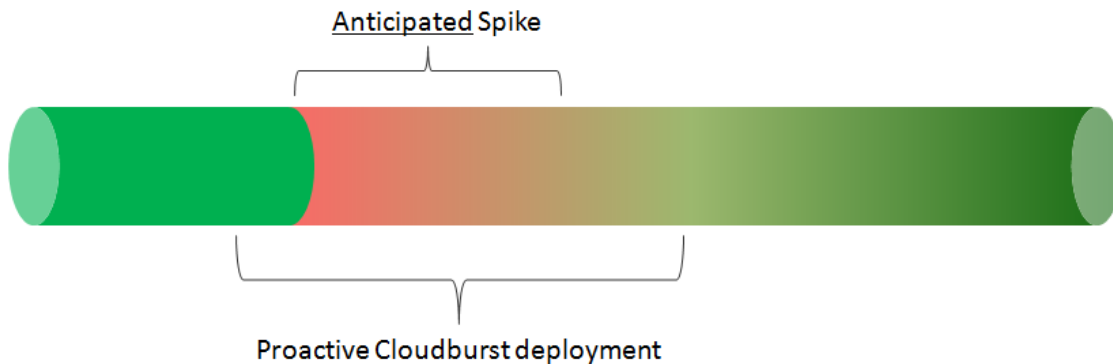
IP Address	Purpose
10.150.24.201	Local Edge external interface (VPN endpoint).
192.168.100.1	Local Edge internal interface (Web – DMZ).
10.1.1.100	vApp Network Edge internal interface (database – backend).
10.150.33.201	Remote Edge external Interface (VPN endpoint).
192.168.0.1	Remote Edge internal interface (web – DMZ extension).

3.4 Triggering Techniques for Cloudburst into the Public vCloud

A cloudburst, in IT terms, is the action of scaling the capacity of an infrastructure to respond to a spike in demand.

For this hybrid scenario, NewCo does not need reaction times on the order of seconds or minutes. NewCo needs overflow flexibility that has predictable patterns. Additionally, NewCo determined that, when they need overflow capacity, they need that capacity only for a few days or weeks. This is because their IT needs are typically associated with marketing campaigns that are programmed upfront and that last for a certain amount of days. Figure 11 illustrates this concept.

Figure 11. Scheduled Burst Period



The triggering of the burst will be process-driven rather than event-driven. NewCo is not going to set up complex infrastructure and application sensors that could automatically trigger the burst event. Instead, NewCo is going to proactively deploy overflow capacity into the public vCloud a few days before the starting day of the marketing campaign so that the hybrid solution is ready to immediately accept the growing number of incoming Web requests.

Though this may not be the most advanced and efficient technique to cloudburst, it is very simple to implement, and is an effective way to meet their requirements.

3.5 Scaling of the Front-End Logic

NewCo wants to take a phased approach with their hybrid vCloud. Rather than investing up front to optimize the entire set of processes associated with scaling the front-end, NewCo is manually deploying additional images and moving them into the public vCloud as part of their scheduled bursting process.

The LOB at NewCo responsible for the new online service maintains a local catalog of templates that they use as a basis for deploying new virtual machines such as those comprising the front-end. This is how they scale the Web instances of this particular online service.

This frequently happens in datacenters when the workload increases and the front-end gets scaled out. This usually involves cloning an existing application instance (or a template that is representative of that particular workload) and making the required application-specific adjustment to join the existing farm that delivers the Web service.

There is nothing in this process that is vCloud-specific—this is what IT administrators have been doing for years. What is unusual in this case is that the additional stacks being deployed will be deployed in an external infrastructure instead of being run locally (and consuming local resources).

3.6 Scaling the Front-End into the Public vCloud Through APIs

At first, NewCo thought they could deploy new instances to the available private vCloud virtual datacenter using the vCloud Director user interface, and later move them into the public vCloud.

NewCo realized that this was not a very efficient way to instantiate similar front-end virtual machines in the public vCloud. They decided to copy the local virtual machine template (that they use as a basis for the front-end instances) into the catalog that is available in the remote virtual datacenter hosted in the public vCloud.

They also realized that doing this manually through the vCloud Director UI could be error prone and would not give them an opportunity to better orchestrate and automate the process. Because of this they decided to implement a few simple scripts to:

- Export the virtual machine template in the catalog of the local organization as an OVF file.
- Upload the template into the virtual datacenter of the remote organization in the public vCloud.
- Add the template to the catalog of the remote organization in the public vCloud.
- Create a single vApp in the remote organization and deploy the required number of front-end instances from the virtual machine template previously uploaded.

All of the required guest operating systems and the application configuration changes for these new instances will be performed manually. The scripts mentioned previously would be run manually as per Section 3.4, Triggering Techniques for Cloudburst into the Public vCloud.

While NewCo runs these scripts manually when required, this lays out the foundation for future potential automation by using an orchestrator product to tie together all these scripts and possibly trigger the burst automatically. While this could be seen as a desired end-state, NewCo realizes all the associated challenges and would rather take a conservative approach on how to burst into the public vCloud. This basic approach is compatible with their requirements.

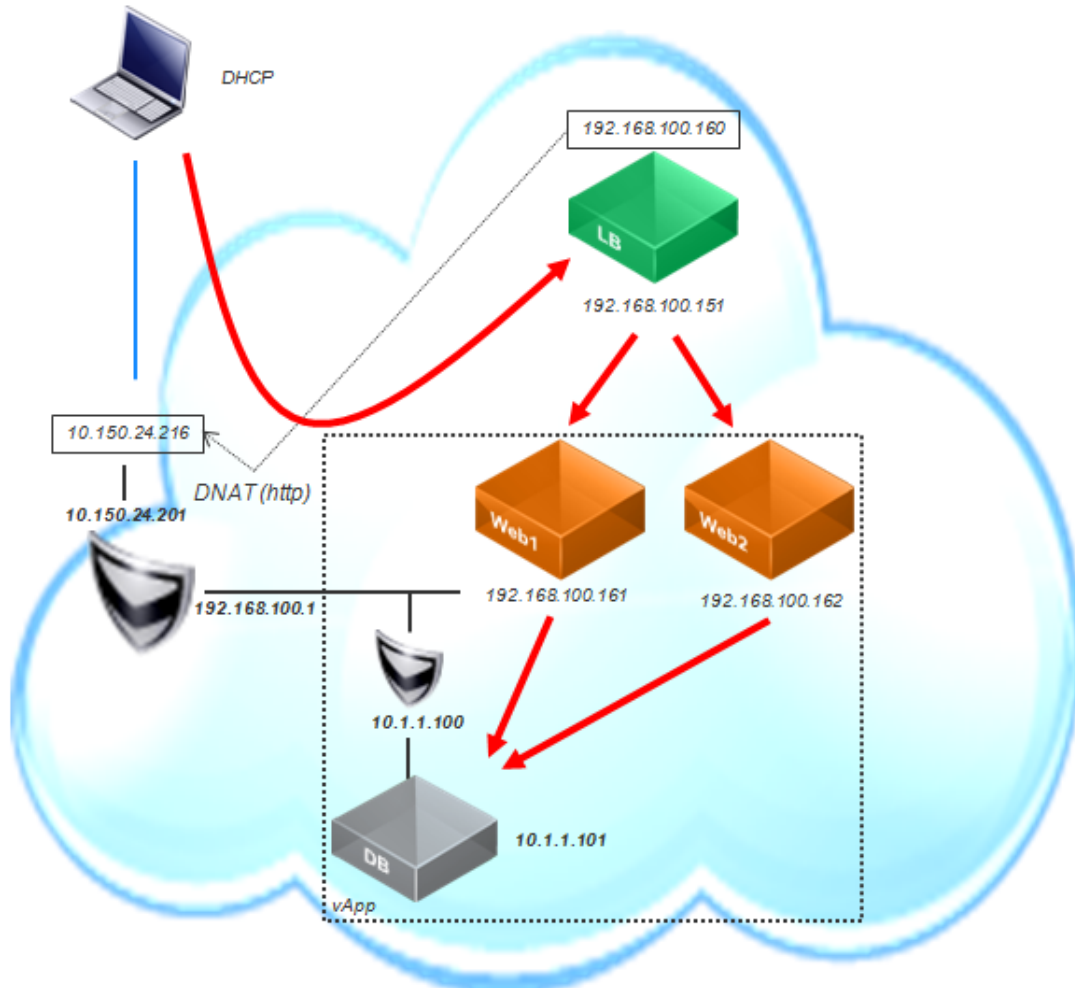
NewCo decided to model their scripts on the public HellovCloud examples provided by VMware as part of the development toolkit.

For more information and background on how to use the Web interface and the APIs to work with vCloud Director organizations and virtual datacenters, see *Consuming a VMware vCloud*.

3.7 Load Balancer Configuration

The load balancing service is the true enabler of the hybrid solution NewCo is building. The typical local implementation is very simple and is comprised of a load balancing (local) service that spreads traffic across a number of (local) front-end servers which, in turn, communicate with the database if and when needed. Figure 12 shows how NewCo deployed this service.

Figure 12. Typical Private vCloud Service Deployment at NewCo

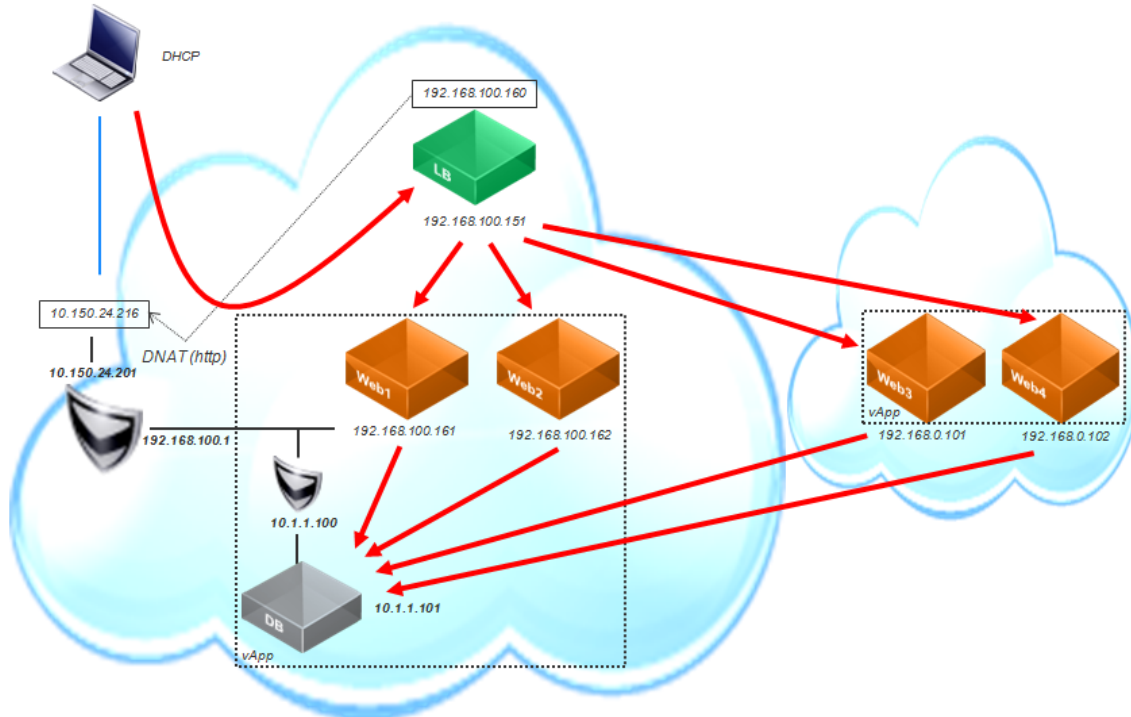


The typical NewCo setup includes a load balancer that sits in the routed organization network in a one-arm configuration that balances the front-end servers. NewCo has sufficient Internet bandwidth so that is not a constraint from a scalability perspective. Similarly, the backend database is not a constraint because they have implemented a layer of in-memory data-replication in the front-end stack that allows the application servers to touch the database only when actually committing transactions (this happens only for a small fraction of all traffic). Because of this heavy decoupling layer, and because of the scalability limitations of the application code, the front-end itself becomes a bottleneck when user traffic increases. Specifically, CPU and memory resources in the front-end layer are the limiting factors to consider when scaling horizontally.

For this reason NewCo is not concerned about creating a global load balancing policy that optimizes bandwidth and locality. They are more concerned with having more available CPU and memory capacity to instantiate new front-end virtual machines when needed.

As a result of their analysis, NewCo decided to implement a very simple configuration that allows their load balancer to reach out to remote instances and balance them as if they were local resources. Figure 13 illustrates this approach.

Figure 13. Load Balancing During the Burst (from a Private vCloud)



The following table summarizes the IP addresses that were used to create this setup.

Table 6. Online Service IP Address Details (Private vCloud and Public vCloud)

IP Address	Purpose
10.150.24.216	NATed entry point for the online service.
192.168.100.160	Entry point for the online service (load balanced virtual IP).
192.168.100.151	Load Balancer IP address.
192.168.100.161	Local front-end #1 IP.
192.168.100.162	Local front-end #2 IP.
192.168.0.101	Remote front-end #1 IP.
192.168.0.102	Remote front-end #2 IP.
10.1.1.101	Backend database.

The load balancer is configured creating a virtual server IP address that balances user requests across the two local front-end servers. During the burst the Load Balancer is reconfigured to add the two remote front-end servers to the list of the targets to balance. The virtual IP address is then NATed externally for the end-user to be able to connect to the Web service.

3.8 Decommissioning Public vCloud Resources

Hybrid vClouds are all about flexibility and a better cost structure. NewCo did not want to size for the peaks. They can overflow capacity into the vCloud when needed and only pay for what they use. To do so, they must decommission the resources that they have instantiated for the burst period. NewCo selected the Pay-As-You-Go subscription model offered by their service provider, and decommissioning of unneeded capacity assumed for this model.

Similar to how NewCo triggers the beginning of the burst process, decommissioning is also a human-driven decision. When IT sees a slowdown in incoming end-user traffic to the online service to the point where the local front-end virtual machines can cope with the traffic, the decommissioning process can begin. This usually happens a few days after the end of the marketing campaign.

NewCo had a choice of turning off the remote instances (paying only for consumed storage) or deleting them (and not having to pay anything to the service provider). Because the front-end stack they are using is frequently updated, and because the scheduled spikes occur only three or four times a year, NewCo decided to delete the remote instances when the spike ends and copy them over when the next burst occurs.