QlikView

Technical White Paper: Clustering QlikView Servers

Clustering QlikView Servers for Resilience and Horizontal Scalability

QlikView 11, Version 1.0

CFG

QlikTech

February 2012

www.qlikview.com

QlikView

Technical White Paper: Clustering QlikView Servers

Contents:

Introduction 3

Why Cluster QlikView Servers? 4

Definitions/Terminology 5

Requirements for a Clustered QlikView Deployment 7

Clustered QlikView Server License Keys 7

Shared Network Storage 7

AccessPoint load-balancing strategies 7

Network Load-Balancer (optional) 9

Building and Installing a QlikView Cluster 11

Summary 15

© 2009 & 2010 QlikTech International AB. All rights reserved. QlikTech, QlikView, Qlik, Q, Simplifying Analysis for Everyone, Power of Simplicity, New Rules, The Uncontrollable Smile and other QlikTech products and services as well as their respective logos are trademarks or registered trademarks of QlikTech International AB. All other company names, products and services used herein are trademarks or registered trademarks of their respective own

Introduction

In this paper we will discuss the architectural and installation requirements and options for building a clustered and resilient QlikView Server deployment.

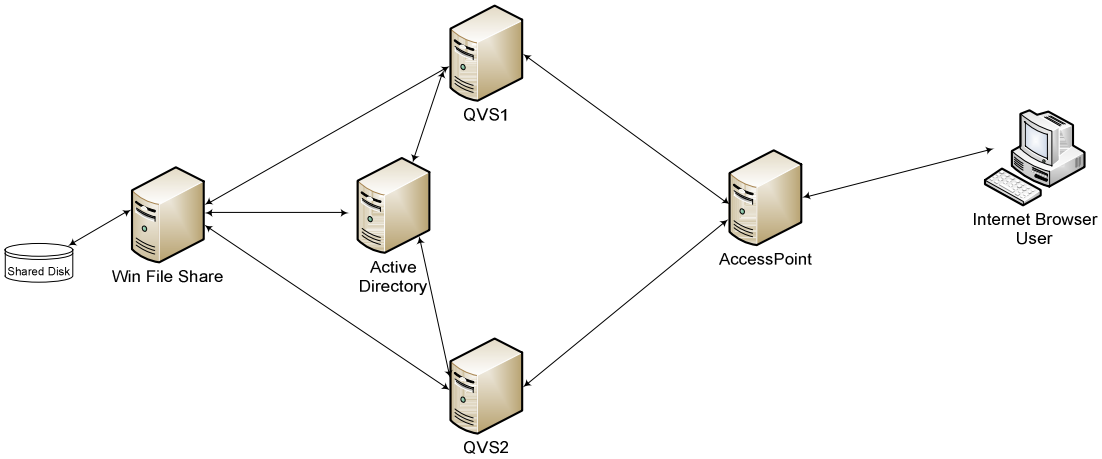


Figure 1: A two server clustered, load-balanced QlikView Server Deployment using AccessPoint (software load-balanced)

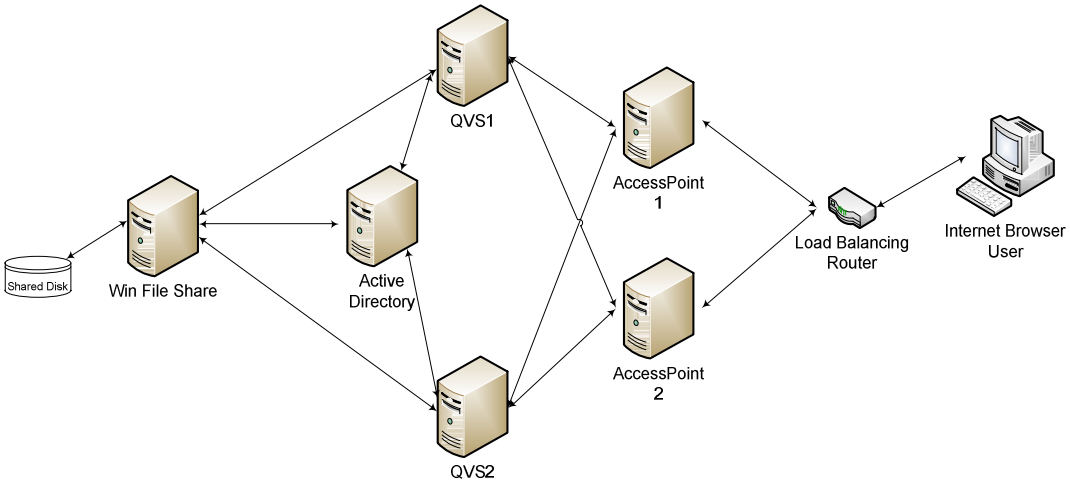


Figure 2: A resilient, two server clustered, load-balanced QlikView Server Deployment using AccessPoint and network load-balancing

QlikView Server load-balancing capabilities are included in the QlikView web portal – AccessPoint. We will also discuss how to make this component resilient using network load-balancing if needed.

Why Cluster QlikView Servers?

Clustering QlikView Servers can allow us to achieve the following objectives:

Horizontal User Scalability

How QlikView scales on a single server is a well understood and documented, see the series of four papers – 'Performance and Scalability', 'Understanding QlikView Hardware', 'Administering QlikView Hardware Tests', and 'QlikView Hardware Sizing Worksheet' for further details. These papers will help you determine what a single server can handle for your data and users.

We can consider adding additional QlikView Server if we need more resource than can be provided by a single server, e.g. if we know that the server we have can support 1,000 concurrent users, but we need to support 2,000 concurrent users, we should consider adding an additional server. In this scenario we could either allocate the first 1,000 user to Server A and the second 1,000 to Server B or if we can cluster the servers so that to the end users there is just one 'world' (in reality it's a single IP address or URL).

Resilience

As the user count on your deployment increases, so does the user's reliance on QlikView, clustering QlikView Servers allows us to build resilience into the deployment. In the case above where we know that a single server can support 1,000 users, to build resilience into the deployment we would consider deploying 3 servers. This allows us to 'lose' a server due to for example, hardware failure and still supports our 2,000 users. Having all 3 servers as active nodes will help reduce response times by not running all servers at 100% utilization and also limit the number of users effected if we 'lose' a node.

Please note that although we are building a resilient QlikView deployment, QlikView does not currently provide any session recovery options. In practice this means that if you 'lose' one of the nodes in your QlikView cluster, users would lose the analysis they are currently performing and have to reconnect to the cluster to resume work. This does not mean that the data within the QlikView application is lost and needs reloaded, as the data is stored in the qvw file on the NAS.

Definitions/Terminology

Cluster:

'A computer cluster is a group of linked computers, working together closely so that in many respects they form a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks. Clusters are usually deployed to improve performance and/or availability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.' [1]

High-availability (HA) clusters:

'High-availability clusters (also known as failover clusters) are implemented primarily for the purpose of improving the availability of services which the cluster provides. They operate by having redundant nodes, which are then used to provide service when system components fail. The most common size for an HA cluster is two nodes, which is the minimum requirement to provide redundancy. HA cluster implementations attempt to manage the redundancy inherent in a cluster to eliminate single points of failure.'[2]

Load-balancing clusters:

'Load-balancing clusters operate by distributing a workload evenly over multiple back end nodes. Typically the cluster will be configured with multiple redundant load-balancing front ends.' [3]

Node

A single QlikView server instance on a server.

Active Node

An Active Node is accepting and processing works.

Passive Node

A Passive Node is inactive, waiting to process work should an active node in the cluster fail.

Network Load Balancer

'In computer networking, load balancing is a technique to spread work between two or more computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, throughput, or response time. Using multiple components with load balancing, instead

[1] http://en.wikipedia.org/wiki/Computer_cluster
[2] http://en.wikipedia.org/wiki/Computer_cluster#High-availability_.28HA.29_clusters
[3] http://en.wikipedia.org/wiki/Computer_cluster#Load-balancing_clusters

of a single component, may increase reliability through redundancy. The balancing service is usually provided by a dedicated program or hardware device (such as a multilayer switch).' [4]

Network Attached Storage

'Network-attached storage (NAS) is file-level computer data storage connected to a computer network providing data access to heterogeneous network clients.' [5]

Storage Area Network

'A storage area network (SAN) is an architecture to attach remote computer storage devices (such as disk arrays, tape libraries and optical jukeboxes) to servers in such a way that, to the operating system, the devices appear as locally attached.' [6]

[4] http://en.wikipedia.org/wiki/Load_balancer
[5] http://en.wikipedia.org/wiki/Network-attached_storage
[6] http://en.wikipedia.org/wiki/Storage_area_network

Requirements for a Clustered QlikView Deployment

There are four high-level requirements for building a clustered QlikView deployment:

1. Clustered QlikView Server license key
2. Shared storage area
3. AccessPoint load-balancing strategies
4. Optionally a network-load balancer for providing full resilience

Clustered QlikView Server License Keys

In a clustered environment the QlikView Server machines are installed with the same license key; this key must be enabled for clustering. You can check this by examining the LEF for the following entry:

NUMBER_OF_CLUSTER_NODES; 2 (number of nodes in your cluster)

Clustered QlikView server's shares configuration and license information between themselves (via the shared storage and via port 14747), so configuration and license management only needs to be performed in once from the QlikView Enterprise Management Console (QEMC) for all nodes.

The servers must be installed on the same network subnet and have a shared root document directory, hence the requirement for a shared network storage. The configuration information is stored in the .pgo files (Persistent Global Objects).

If your servers fail to start or reset after 10 minutes check for the LEF entry above, this is usually an indication of multiple non-clustered servers with the same license key being used.

Shared Network Storage

This is required not only for the .pgo file mentioned above but also for storage of QlikView applications that are required on the cluster. This also enables collaborative objects to be shared across the nodes in the cluster (.shared files).

This is the 'Shared Disk' located on the left hand side of Figure 1 and 2 above. A clustered QlikView deployments utilizes a Windows Server based machine

This is required not only for the .pgo file mentioned above but also for storage of QlikView applications that are required on the cluster. This also enables collaborative objects to be shared across the nodes in the cluster (.shared files).

This is the 'Shared Disk' located on the left hand side of Figure 1 and 2 above. A clustered QlikView deployments utilises a Windows Server based machine

QlikView requires the storage of documents (qvw's), .pgo, .meta and .shared files to be hosted on a Windows Based File Share. Hosting files on any other type of system is unsupported and can create an unstable QVS cluster where CAL's disappear and QVS's stall. QlikView supports the use of a SAN (NetApp, EMC, etc) with a QlikView Server provided it is mounted to a Windows Server (2003, 2008) and then shared from that Windows server.

AccessPoint load-balancing strategies

QlikView AccessPoint supports two load balancing strategies:

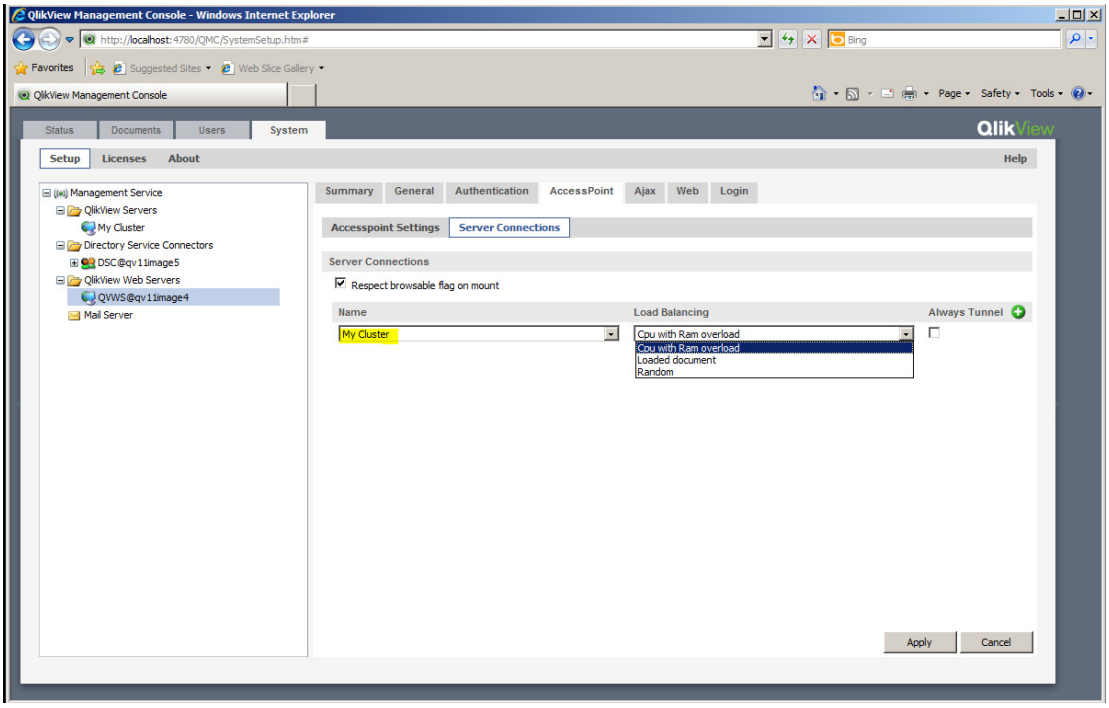
- Random – the default setting, a round-robin type strategy ideal for most users as the session will be distributed across all nodes in the cluster.
- LoadedDocument – used when you want sessions for the same document to be routed to the same server. This strategy is designed for deployments where you have more documents that a single node in your cluster could handle. AccessPoint makes a decision based on if the document is already loaded and the available RAM on the server.

For QlikView 9 these are set in the config.xml file on the server where AccessPoint is installed, usually under C:\Documents and Settings\All Users\Application Data\QlikTech\QvWebServer for Windows 2003 Server or C:\ProgramData\QlikTech\QvWebServer on Windows 2008 Server.

The following section of the config.xml controls the load-balancing and is configured to Random by default once you have created the cluster in the QlikView Enterprise Management Console (QEMC), see later in this document for details on configuring a cluster in the QEMC.

```
<AddCluster>
     <Name>MyCluster</Name>
     <LoadBalancing>Random</LoadBalancing>
     <AddQvs>
       <Machine>qvs1</Machine>
       <Port>4747</Port>
       <LinkMachineName>qvs1</LinkMachineName>
       <AlwaysTunnel>False</AlwaysTunnel>
       <ProxyMode>True</ProxyMode>
       <Weight>1</Weight>
       <Username />
       <Password>Encrypted=DxdCGMWfOwU=</Password>
     </AddQvs>
     <AddQvs>
       <Machine>qvs2</Machine>
       <Port>4747</Port>
       <LinkMachineName>qvs2</LinkMachineName>
       <AlwaysTunnel>False</AlwaysTunnel>
       <ProxyMode>True</ProxyMode>
       <Weight>1</Weight>
       <Username />
       <Password>Encrypted=DxdCGMWfOwU=</Password>
     </AddQvs>
   </AddCluster>
```

QlikView load-balancing strategy can be set from the QEMC, under System, Setup, QlikView Web Servers, select your webserver (either IIS or QVWS) on the AccessPoint tab:



Network Load-Balancer (optional)

This component provides the resilience for AccessPoint, routing the sessions to an available AccessPoint server.

There are several requirements on the load-balancer:

1. Supports 'sticky sessions' – this ensure a users session persists on the same node within the cluster, usually by using a cookie to achieve this.
2. Availability – the load balancer checks the availability of the AccessPoint web server and QlikView servers.
3. Some form of load balancing algorithm to determine which server is least loaded.

Sticky Sessions

The requirement is for the user's session to be routed consistently to the same server. Methods of doing this vary from device to device – please refer to your load-balancer vendor's documentation on what options are available.

Availability Checking

A special web page is available on the AccessPoint that allows for automated checking of the system status:

http://myAccessPoint/QvAjaxZfc/QvsStatus.aspx

This page returns an http status code of 200 if the AccessPoint and at least one QVS in the cluster is responding, any other status code returned by this page should be considered an error. Common errors from this page:

404 – The AccessPoint is unable to respond, check the Web Server.

503 – No QVS's responded to the AccessPoint and therefore it cannot service user requests.

If you view the above page in a web browser you will see the status of your QVS cluster:

Name	Host	Status	Reason
MyCluster	qvs1	Up	
MyCluster	qvs2	Up	

Load-Balancing Strategies

Here we need to consider how the load-balancing router will allocate sessions to the nodes within the cluster, several different strategies are possible:

Round-Robin

The Load Balancer sends each session to the next available server; this is a fairly rudimentary load-balancing algorithm.

Session Counts

This uses the load-balancer to keep a running count of the number of session on each AccessPoint and ensuring that they are equal number of sessions on each node.

Building and Installing a QlikView Cluster

Using the QlikView Enterprise Management Console (QEMC) follow the steps below to configure and activate your QVS cluster.

1. Install and license the 1st QVS in the cluster, this will be your “master" QVS.

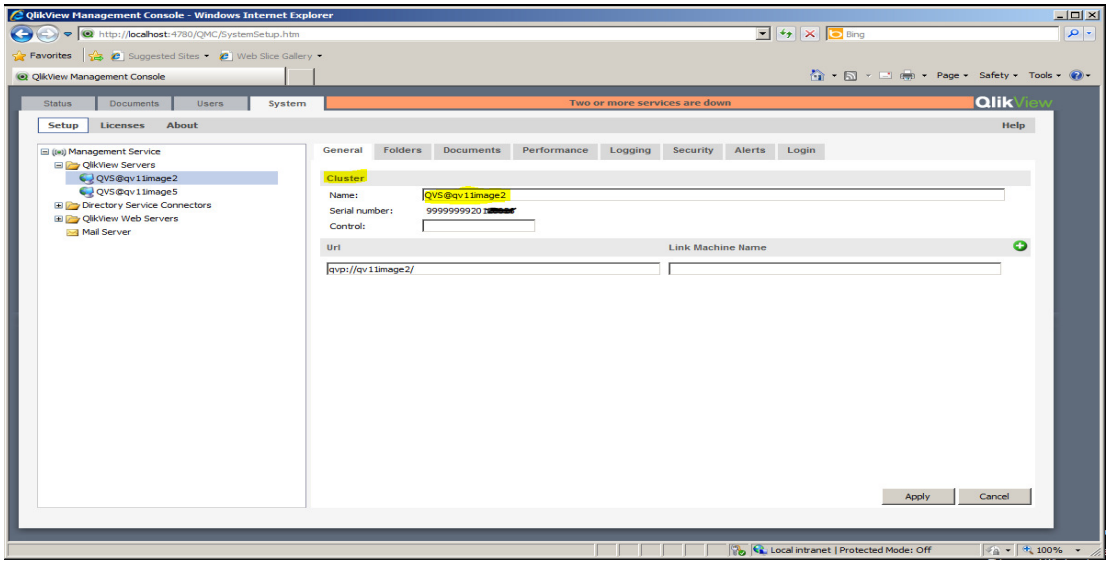


Figure 3: QVS@qv11image2 configured as a “master” QVS

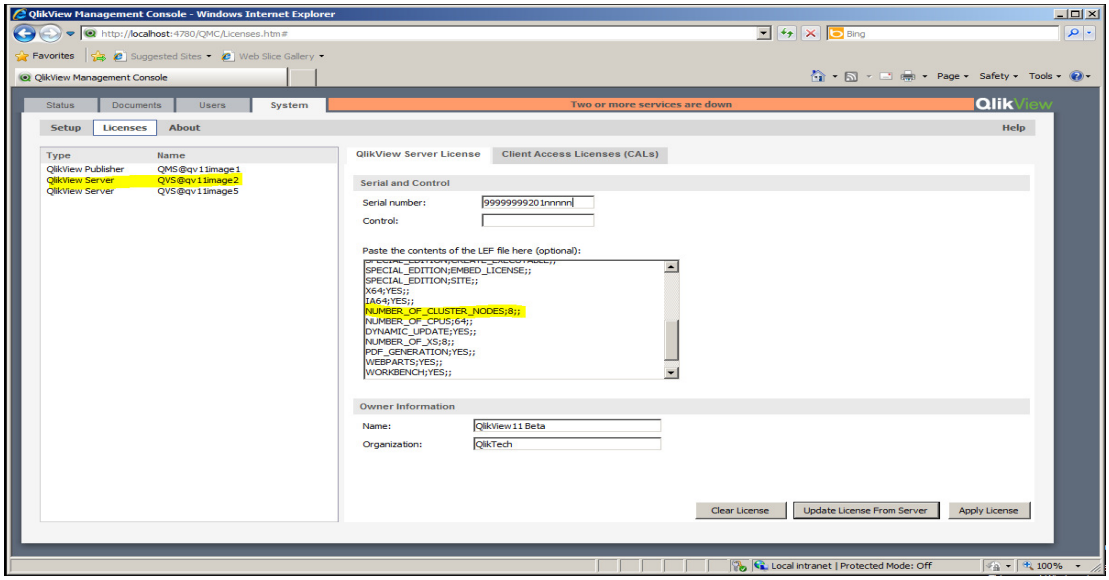


Figure 4: Installation of a “Cluster” license.

2. Configure the document directory to point to a directory that all QVS in the cluster can access on the NAS.

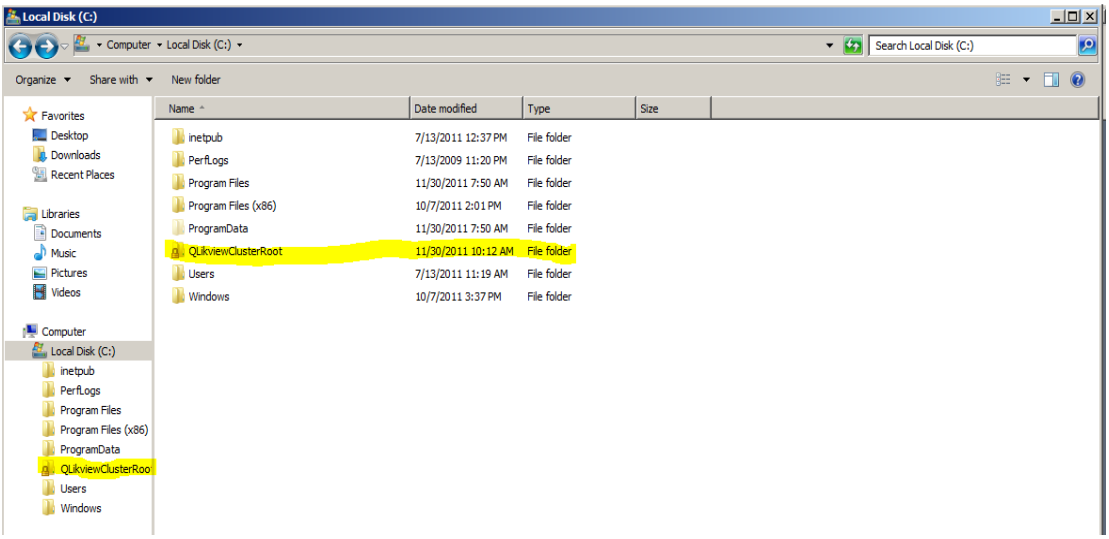


Figure 5: Windows Explorer showing QlikViewClusterRoot as a shared folder.

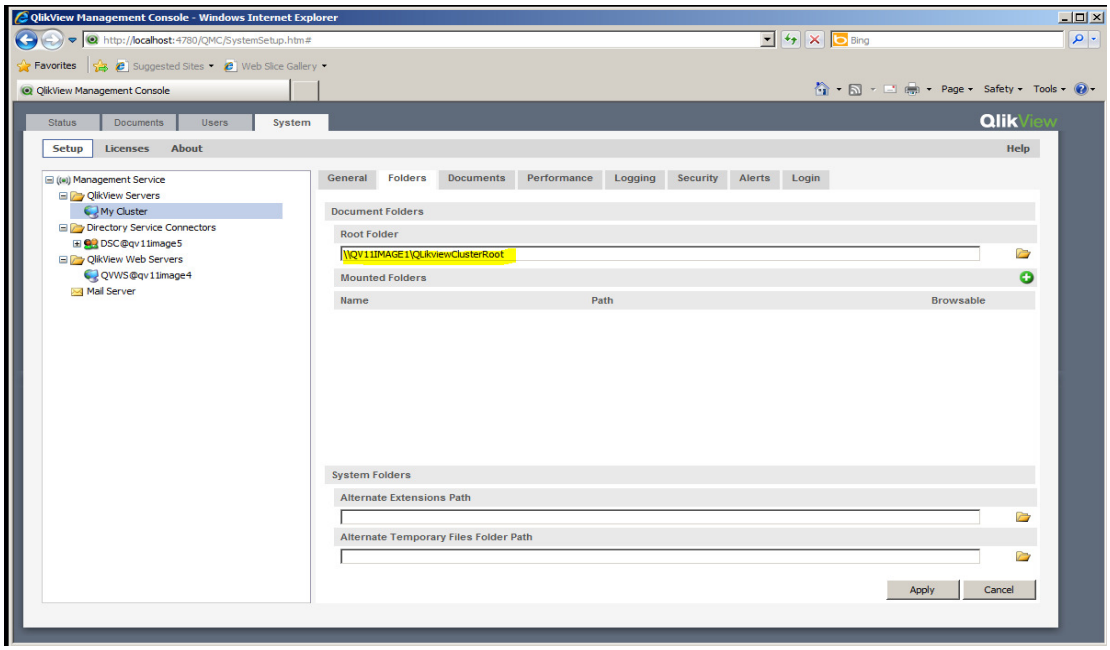


Figure 6: \\QV11IMAGE1\QlikViewClusterRoot configured as a folder in QlikView.

3. Install the next QVS in the cluster.
4. Ensure all QlikView services are running as local administrators and also have membership of the 'QlikView Administrators' local group.
5. In QVS settings on the cluster tab add the control number to your license and the address to the 2nd QVS in the cluster.
6. For usability reason you can now rename the cluster on the General tab for the QVS.
7. Repeat step 3-5 for the other QVS nodes in the cluster

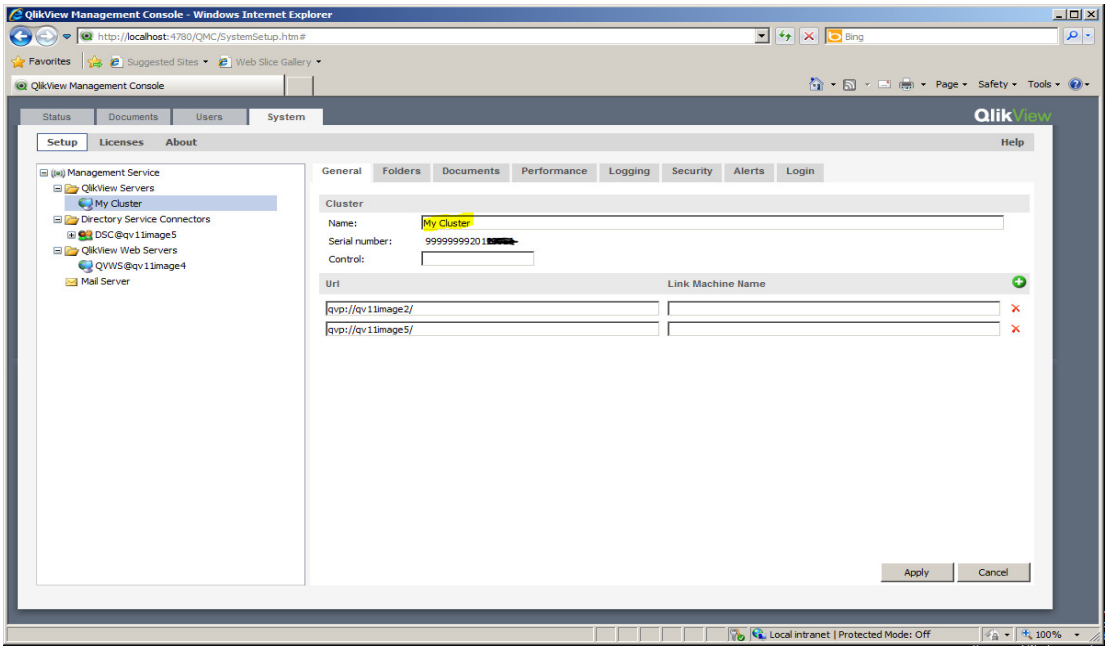


Figure 7: A two node QVS Cluster configured in the QEMC

8. In the Access Point setting, make sure that your cluster is selected in Server Connections

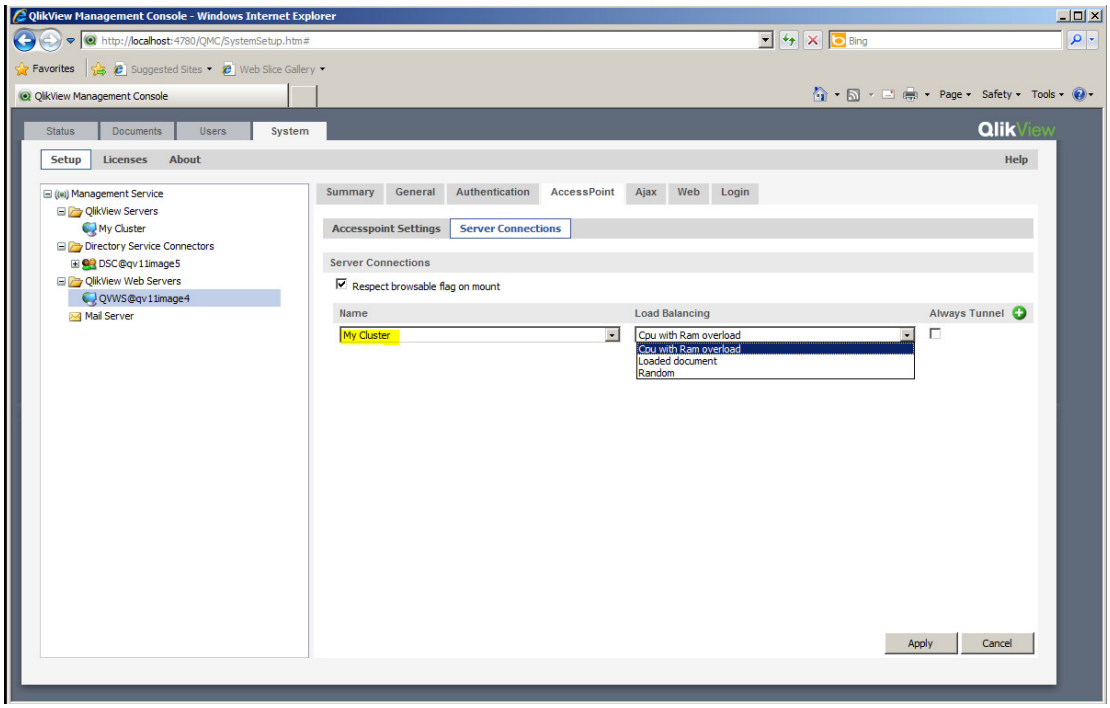


Figure 8: AccessPoint configured to use 'My Cluster' cluster of QVS's

9. The cluster is now configured and ready to use.

Summary

Hopefully by now you will have an understanding of the infrastructure requirements for clustering QlikView servers and can start planning your deployment.

As a recap these are the things to consider:

- Why am I clustering – resilience or additional QlikView Server resources or both?
- How many QlikView Servers will I cluster?
- Do I have a 'cluster enabled' QlikView Server Key?
    - Does it have the relevant number of servers set?
- Shared Storage infrastructure in place?
- Which AccessPoint load-balancing strategy do I want to use?
- What hardware load-balancing strategy do I want to use?

If you have further questions or require assistance in building your QlikView Server cluster please contact your local QlikTech office for assistance from our Professional Services Team.