QlikView

QLIKVIEW AND THE CLOUD

A QlikView Technology White Paper

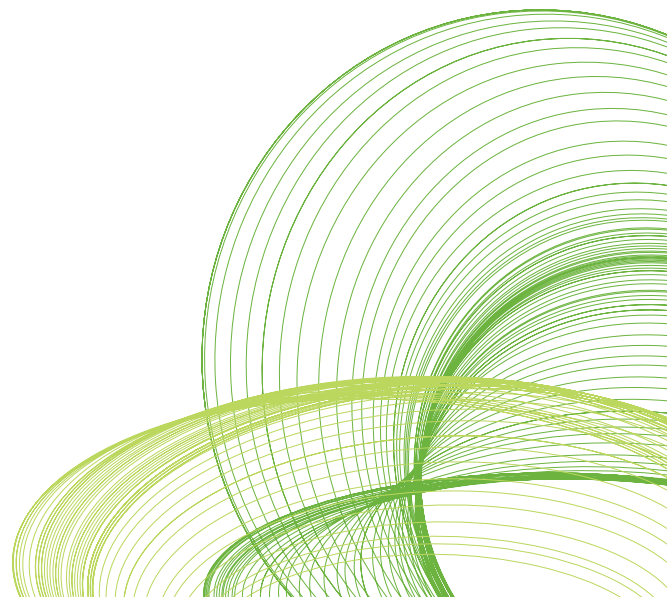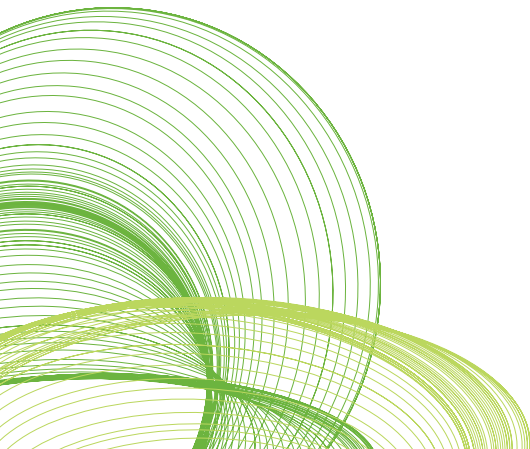July 2012

qlikview.com

QlikView

Table of Contents

QlikView

Overview

The demand for and implementation of virtualization technologies internally within organizations during the last few years has revealed significant savings and reduced risk for many businesses and IT teams. Cloud is the next logical step beyond internal virtualization. Improvements in network bandwidth and cultural demand from users to be able to take their services and content with them wherever they go is making organizations reconsider how they deliver their IT services to their users.

The use of cloud computing and software as a service (SaaS) represents a huge growth area in the IT industry, with many analysts predicting growth. According to Forrester Research, the SaaS space will grow significantly during the next five years to reach total revenues of 92.8 billion USD by 2016 — accounting for roughly 26% of the total packaged software market. [1]

The range of benefits offered by using cloud services and the maturity of the key cloud vendors is driving adoption. Benefits of the cloud include fast implementation, user-friendly experience, IT infrastructure support not required, ease of leveraging cloud-based data, and automatic, seamless upgrades.

And when cloud offerings are licensed to customers on a term basis (e.g., per month or per year), this can result in cost savings, which is a key motivation for adoption. Delivering solutions as a service offers easy opt-in/opt-out and enables customers to turn what were capital expenses into operating expenses.

Many QlikView customers are now looking to the cloud deployment and/or SaaS licensing to provide Business Discovery to their users. This QlikView Technology White Paper discusses a variety of cloud and SaaS options. In addition, we provide a deeper look at two significant cloud services, Amazon EC2 and Microsoft Azure.

WHO IS THIS PAPER FOR?

This paper is written for anyone who has an interest in QlikView's role in the ever-growing topic of business technologies utilizing the capabilities that cloud-based infrastructure offers. It will be of most interest to two primary groups of readers:

- QlikView customers who are interested in understanding the options for hosting their own QlikView deployments off-premise and in the cloud, typically via infrastructure as a service

- QlikView OEM partners who are interested in understanding the options for licensing their QlikView-built applications via software as a service (SaaS) model for their customers.

Finally, the paper is geared towards a more technical audience; however it is not restricted to just this group of people as it contains explanations about topics that are increasingly part of the vocabulary of anyone involved in purchasing, deploying and using business software.

Cloud Terminology and QlikView

Cloud as an overall concept refers to an off premise software deployment model in which a service provider delivers some combination of hardware, software and services to their customers. The cloud may be public, like Amazon EC2 or Microsoft Azure, or it may be private. The responsibility for running and maintaining a particular service is transferred to a 3rd party and the degree to which this responsibility is transferred depends on the type of cloud service offered.

Figure 1: QlikView can be deployed anywhere

QLIKVIEW ON PREMISE

Apps

App Platform

Data Sources

QLIKVIEW IN THE CLOUD
(public or private)

Apps

Data Sources

App Platform
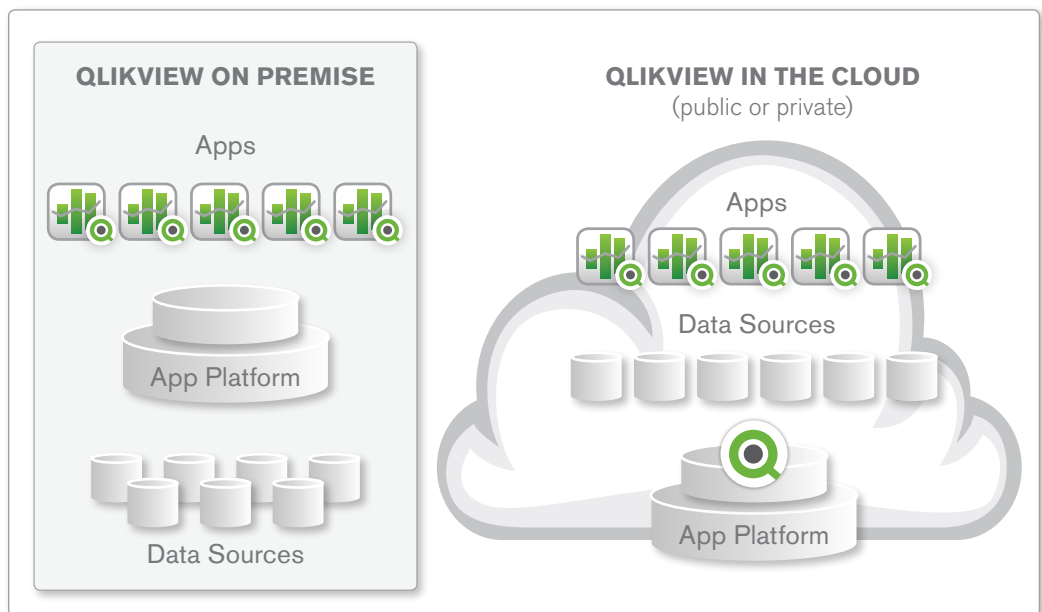
Below are some of the common cloud services and terminology offered under the banner of cloud and their relevance to QlikView:

SOFTWARE AS A SERVICE - SAAS

SaaS is a term licensing model; customers pay for the use of it for a specified period of time. In many cases, the software application is fully developed, maintained and provided as a service to multiple customers. Often, several customers use the exact same application but typically see just their own data. One well-known example of a SaaS offering is the CRM vendor Salesforce.com®.

Typically a SaaS system is provided as-is and while the customer has some ability to configure it for their use it cannot be heavily customized or modified by the customer. Similarly the SaaS vendor can upgrade or change the way the software works and so all of their customers need to align with this. In a SaaS model data entered into the system will be stored off premise so customers must trust and ensure that the vendor's safeguards around their data are in place.

QLIKVIEW AND SAAS

SaaS is probably the most common term used with cloud computing and can be misinterpreted in its relation to QlikView. With the SaaS model all users of the service receive the same application experience. With QlikView this would mean all users would view the same QlikView apps but perhaps the data they see within those applications would be different.

Most QlikView customers want to create their own QlikView apps on their own data and distribute those to their users, so by that fact a SaaS model is not always appropriate. This need can be served better by some of the other cloud models shown below.

QlikView does not directly provide a SaaS service offering: however many QlikView partners do offer QlikView-built applications as a SaaS offering to their own customers. Often, they have identified a specific niche for a QlikView application and provide access to that as a subscription service. Some examples of customers offering QlikView is this way are included in the blog post "SaaS BI: It's All About the Apps":

- **CONTEXT** provides ICT (information and communications technology) channel sales and price tracking

- **IFR Monitoring** delivers market research for retailers

- **InContact** delivers contact center software

- **Infostrada Sports Group** delivers sports statistics

- **Kenexa** delivers workforce management solutions

- **Rosslyn Analytics** offers cloud-based spend analysis apps for procurement, finance, accounts payable, and IT teams

- **SynerTrade** delivers purchasing software and consulting

- **Xerox's ACS Financial & Securities Services** delivers a sales tracking app for mutual fund companies

These partners offer domain-specific or industry-specific apps in areas like market research, workforce management, purchasing and spend management, mutual fund or mortgage sales tracking, channel management, and others.

INFRASTRUCTURE AS A SERVICE – IAAS

IaaS is the term used to provide server instances as a service on demand. For example one can create a virtual Windows Server on the fly, pay by the hour while one needs it and then turn it off whenever one wants to. The server itself can be used for any purpose.

Together with these servers are a range of other components such as network, storage and security services to form up the overall infrastructure service. Example of IaaS vendors are Amazon with their EC2 service and Rackspace but there are many more.

With IaaS the vendor maintains and takes the risk of operating the infrastructure and ensuring that the service is kept online and can handle peaks and troughs in demand. For the customer the key benefit is the ability to add boxes on demand without fixed term costs and to be agile with how many boxes they use. For example if a peak in web traffic is expected during a busy period, additional servers can be added to the infrastructure and then simply turned off when the demand lowers. Servers can be made available within minutes and with a variety of configurations so business can be very agile with infrastructure needs compared to the often long buying cycles applied to internal hardware purchases.

With IaaS the customer is typically responsible for maintaining the operating system and the software they run on it. So compared to the SaaS model there is still a requirement for some degree of systems administration but it is with the customers control to do it when they choose to.

QLIKVIEW AND IAAS

IaaS is a common cloud model used by QlikView customers; indeed QlikTech's own demo.qlikview.com and other services are run from the IaaS model offered by Amazon EC2. This model allows QlikView customers to deploy infrastructure suitable for their environment and have full control over what QlikView applications they host in the environment and how they configure it.

QlikView Server supports the Windows 2003 and 2008 operating systems and so when these are used as the operating systems provided by the IaaS vendor, customers can be sure they are running on a supported platform.

PLATFORM AS A SERVICE – PAAS

Platform as a Service is a different approach to the above models in that its primary intention is neither
to provide a specific common software service nor to provide solely infrastructure. The model is intended to provide an application platform on which a customer can host applications that they build themselves but without having to be concerned about or maintain the platform that hosts it.

Examples of this type of service include Microsoft Azure and the Google Application Engine.

With this model a customer builds an application using, for example, .net with a SQL database. This application is then packaged and submitted to the platform. The platform vendor is then responsible for provisioning resources to host the application and ensuring that any operating systems, such as Windows, are maintained and up to date and available.

The advantage of this model is that it allows customers the flexibility they want but with a very minimal need to maintain any infrastructure at all. With this model, however, the customer has to use the technologies provided by the vendor's platform.

QLIKVIEW AND PAAS

Very often it is this model that people tend to think of – and somewhat incorrectly -- when they think about "QlikView via SaaS." If people ask whether QlikView is available via a SaaS model, they are typically thinking about QlikView servers in the cloud that they could sign up for and use to create their own QlikView apps, rather than accessing prebuilt and customized applications that our OEM partners mentioned earlier typically provide.

With QlikView in a PaaS model, a customer could implement QlikView onto a hosted PaaS vendor's technology. For this type of implementation, QlikView requires a Windows platform to run on. One such PaaS option is Microsoft's Windows Azure. It is possible to package the QlikView product along with customer - or partner-built apps and deploy onto the Azure platform and so enable customers to be free of the infrastructure (i.e. hardware, network and OS) management.

CLOUD STORAGE & DATA

Most of the cloud vendors provide some method of storage as part of their offering. Cloud storage allows customers to store data in a secure and resilient way for a number of applications and provide high bandwidth access to the data. Customers often pay for the volume they store and then for the amount of data downloaded.

Cloud storage is very useful when a customer needs to distribute high volumes of data to a wide user base without needing to implement high bandwidth access to its own network. QlikTech uses this approach to distribute large software installation files to its customer base as do many other software vendors.

Many of the cloud storage mechanisms have programmable API to allow them to upload and manage content. While software such as QlikView isn't applicable to cloud storage it is a component within the cloud stack than can be used to get data into and out of the cloud, so it is worth noting here. Also, some QlikView SaaS OEM partners provide capabilities for uploading data into the cloud, where it is then typically enhanced and combined with other cloud data.

Approaching QlikView in the Cloud

This section discusses some of the broad topics to address when hosting QlikView in the cloud and looks at ways some of the cloud vendors can support QlikView implementations. This section is most relevant for a) QlikView customers who wish to understand the various hosting and infrastructure options for their QlikView deployments via an IaaS or PaaS model and b) OEM partners who wish to understand the options for building their own QlikView applications and making them available via a SaaS model.

KEY ASPECTS OF HOSTING QLIKVIEW IN THE CLOUD

Regardless of the technology or software, use of cloud based services is often something new to an organization and so there is to some extent some uncertainty over the risks involved in using them. In some cases use of the cloud means that an organization's data has to leave the confines of the internal corporate network and be located in the cloud. This means that an appropriate architecture and supporting processes have to be put in place to ensure that data is secured and accessed only by those permitted to see it. In addition the mechanism for moving what could be large volumes of data must be considered. In other cases, the data is already in the cloud.

Typically when organizations want to make a service available to users outside of their corporate network (which could be both employees from home or fully external parties), attention is paid to the security of the infrastructure surrounding an application as well as the security of the application itself, such as identifying who users are and what they can do.

From an infrastructure perspective in a traditional approach to QlikView deployment, organizations may make use of specialist techniques and components such as DMZs (DeMilitarized Zones), proxies and other hardware devices. Most of these can be replicated with the features offered by cloud vendors.

This paper will discuss security in relation to QlikView but organizations looking to host in the cloud should also consider best practices for security in the cloud as a general topic. Most IaaS and PaaS vendors provide information on this topic for their offering.

With regard to data, in an on-premises deployment, QlikView may extract data from source systems. When using QlikView in the cloud, consideration has to be paid to how data can be accessed from source systems which may be internal to the corporate network and how much data is moving. Options for this are discussed below.

Security Integration

The objective with QlikView security is to be able to securely identify users and what they are permitted to do. To this end QlikView can integrate with a range of security mechanisms and this flexibility can allow customers to choose which approach works best for them in a cloud environment.

With an on-premise deployment of QlikView a large percentage of customers will use the Windows-based security mechanisms based on Active Directory. This is an out of the box configuration that requires little to no set up.

When a cloud server is used it is typically not part of the Active Directory and so cannot natively use those Windows accounts. While it is possible to join a cloud server to a customer's internal domain we will discuss the alternatives below. The options available depend on what the customer has available in terms of their infrastructure.

All of these approaches are common methods used in QlikView and more details can be found in the QlikView Security Overview Technology White Paper. [2]

SSO / HTTP HEADER / REVERSE PROXY

Many customers that have a Single Sign-On (SSO) system in house (such as Tivoli Access Manager or SiteMinder) will have the capability to use a reverse proxy within their environment. Essentially what this does is it centrally controls both the sign in mechanism and the point of access to the content that users want to get to.

Here is how it typically works…
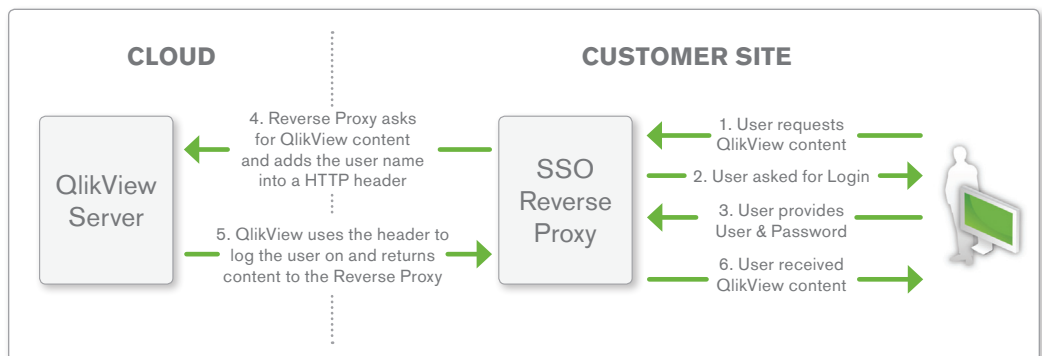
In this kind of deployment there is no customization required for QlikView, just a simple configuration; however it does require that the customer has in place an SSO system that can make use of reverse proxy.

In this type of setup the only device that ever physically touches the cloud server is the reverse proxy. No users will ever directly touch the cloud server or even know that it is a cloud server. This mechanism could serve both internal and external users and is an excellent approach for conforming to what can be the standard way of integrating security within an organization.

SAML / FEDERATED SECURITY / SECURE TOKENS

There are a number of different security systems that can make use of secure tokens to sign users into a range of systems. There are several different standards and sets of terminology around this approach such as SAML (Security Assertion Markup Language) and federated security such Active Directory Federation Services. Although each is different the approach from a QlikView perspective is similar.
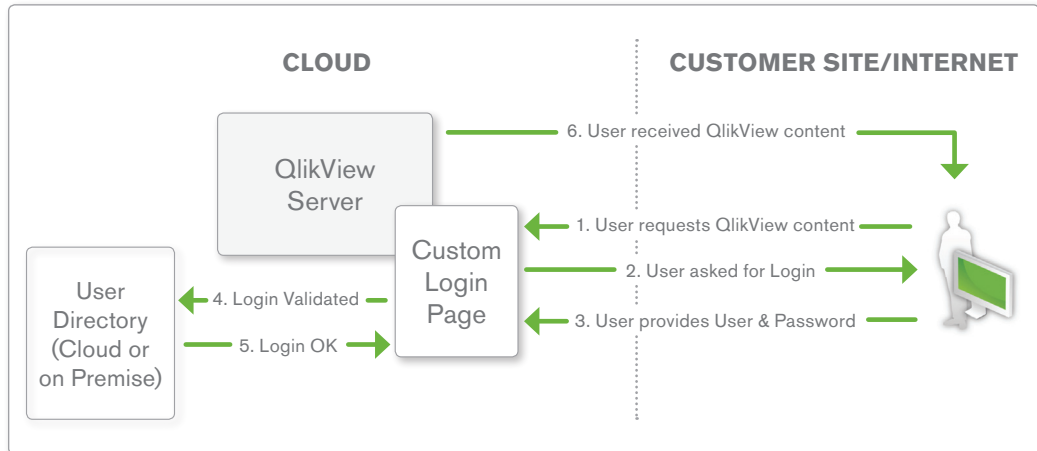
Here is how it typically works…
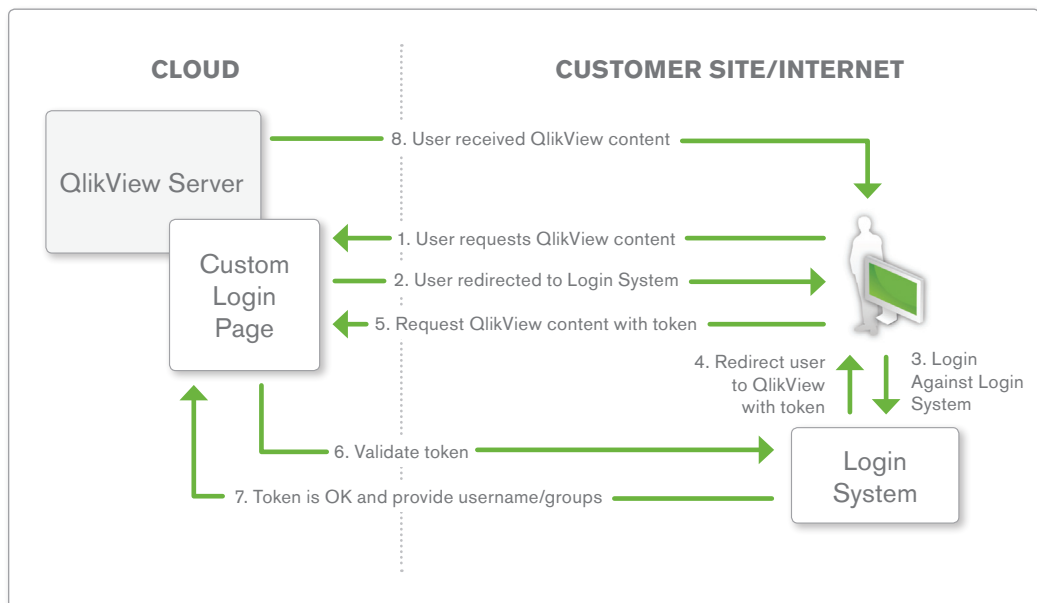
In this approach, due to the level of integration required, a custom login page can be created to use the security API in QlikView. Although this mechanism is relatively simple to use it does require some knowledge of programming and this configuration is not 'out of the box'. There are examples on how to implement this approach available on the QlikView Community. [3] The advantage of this approach is again to conform to an organization's standard way for securing services. It again does require that an organization has in place a security system that offers this kind of functionality but QlikView has an approach for integrating with a range of vendors' solutions.

With Active Directory Federation Services (ADFS) it is possible for users to seamlessly log in to a non-domain cloud server without being prompted to log in as their internal Windows credentials are used during the process of logging in. This gives an excellent user experience and ADFS is often implemented within organizations alongside their regular Active Directory which means there is no requirement for an additional SSO product or set of users.

CUSTOM LOGIN / USER DIRECTORY

In some cases there is no security system to integrate with beyond perhaps a known list of user IDs and passwords. In these cases a straightforward login page can be created in a similar way to the method previously mentioned to ask for, and validate, a user ID and password before logging them into QlikView.

Here is how it typically works:

The user directory could be anything, anywhere as long as it is accessible from the cloud. It could be on customer premises or in the cloud; ideally it should be an existing set of user details but could be created just for QlikView. Directory examples include:

- Active Directory

- LDAP

- SSO systems

- Databases (SQL, Oracle etc)

- Web based systems such as Windows Live, Google accounts or Facebook login

Although there is no specific security product required, this approach will require a custom login page and can be created to use the security API in QlikView. Although this mechanism is relatively simple to use, it does require some knowledge of programming and this configuration is not out-of-the-box.

Data Update

Typically, new data needs to be loaded into QlikView applications on a regular basis to suit user requirements. This could be hourly, daily or weekly for example. When working in a cloud scenario it can be that the data is inside the corporate network and so there are a number of ways that QlikView can be configured to get access to this.

DIRECT ACCESS

In this model a port on a corporate firewall is opened between the cloud server and data source inside the network. QlikView then reloads apps on a schedule and that reload effort occurs in the cloud.
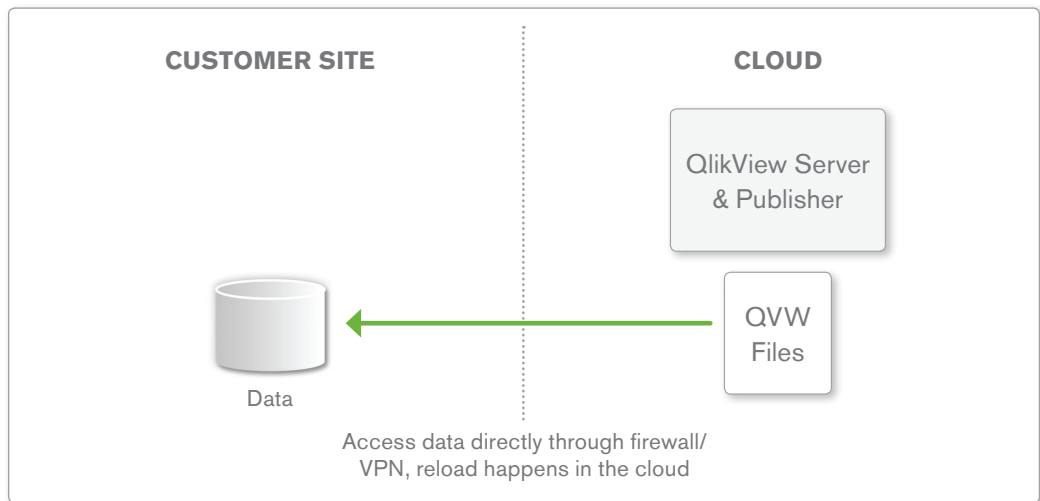
When using direct access it is advisable to apply security to this by using technologies such as a VPN or SSL. If large data volumes are being used or if the network speed is slow then this model may not be suitable. Techniques such as incremental load may help to alleviate this.

DATA SYNC / PUSH

Rather than reloading QlikView from a remote source, the data sync/push model has data being synchronized from the source system out to the cloud server and reloaded from there. This push could take the form of a simple FTP of files or a sync of a database such as SQL Server.
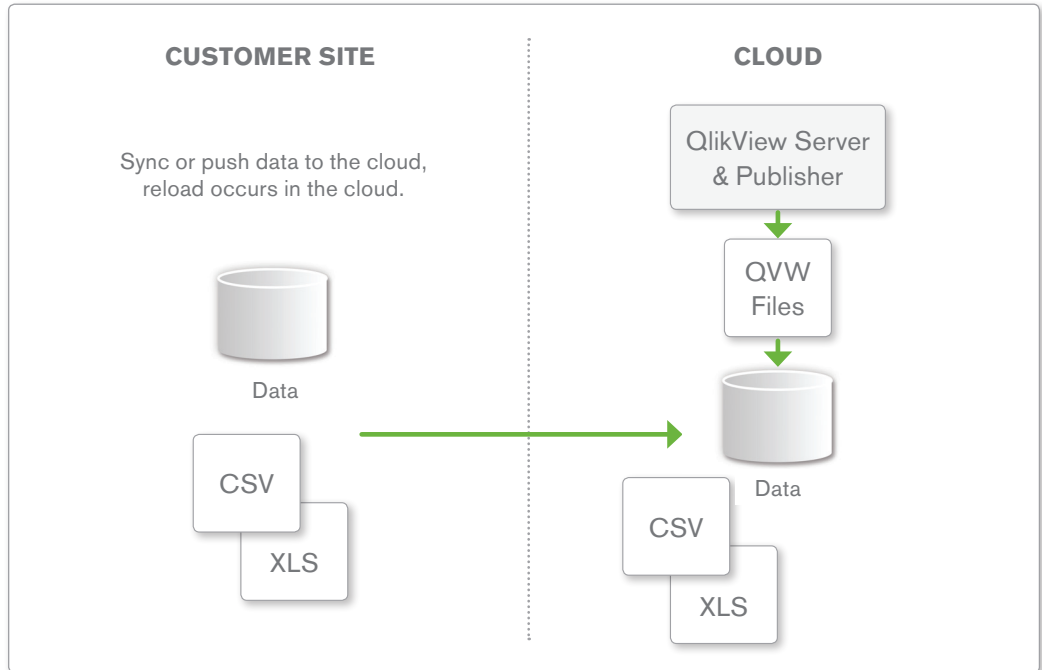

This approach relies on a task or process occurring inside the corporate network to push the data out. This would ideally be automated but for scenarios where data update done infrequently this could also be done manually.

PUBLISH COMPLETE QVW FILES

In this scenario the QlikView Publisher component is hosted inside the corporate network where QlikView applications are reloaded before being distributed to the cloud-based QlikView Server. This model requires no sync process or access through firewalls to data.

An additional advantage of this approach is that the QVW (QlikView data file) contains the data in a highly compressed form, reducing the amount of data to be distributed. The disadvantage of this model is that the Publisher component must be hosted on the internal network and avoiding internal hosting could be a key motivation for using cloud. However this could be a virtual machine internally.
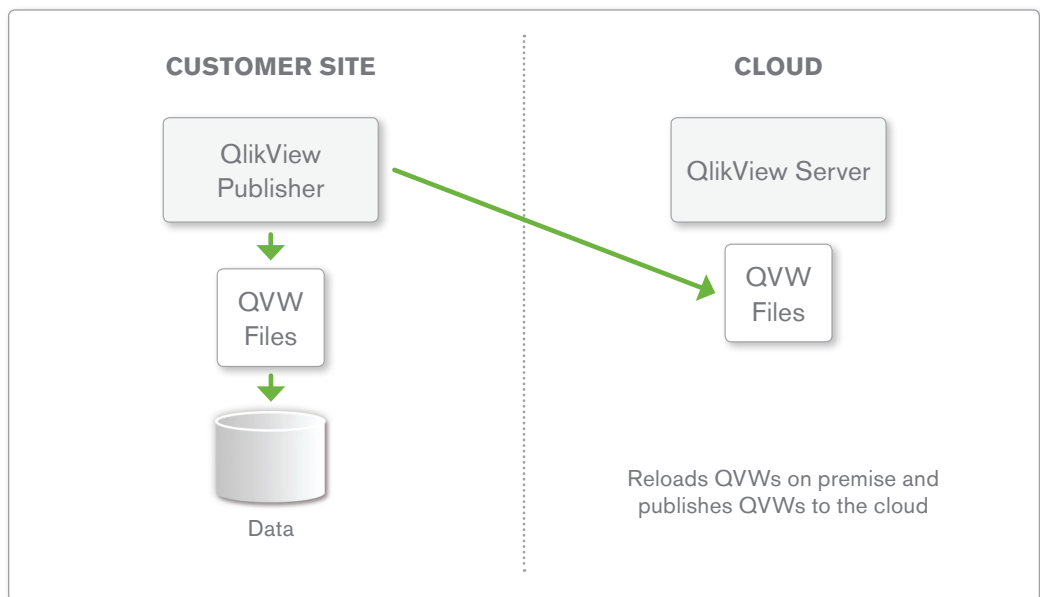
CLOUD-BASED DATA

For scenarios where data is cloud-based to begin with then QlikView in the cloud is a perfect fit. Data can be reloaded directly from the source system. In fact, many of QlikTech's OEM partners employ this model as it assumes that all of the concerns about data security, location and access have been already addressed with the transactional system and makes incorporating an analytics solution in the cloud much more straightforward from a compliance and agreement point of view. A well-known example of cloud-based data is Salesforce.com for which QlikView provides a data connector. In addition security integration with cloud based systems can be achieved.

CLOUD

QlikView Server

QVW Files → Cloud Source Data
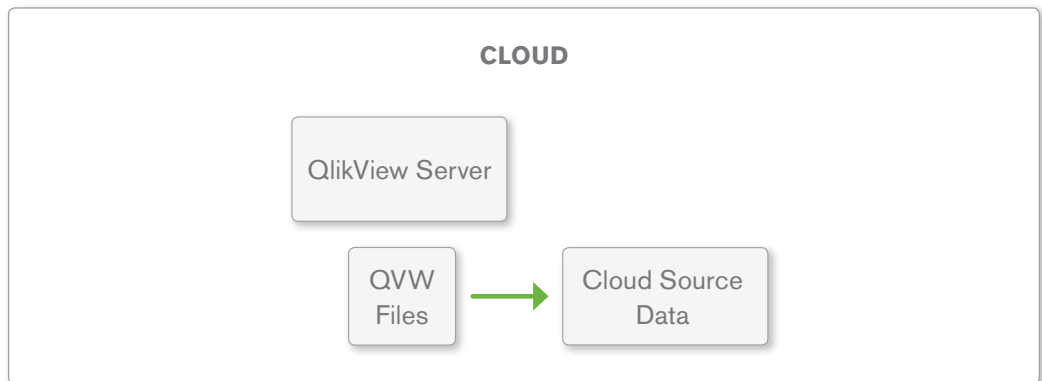
Scalability in the Cloud

CLOUD, VIRTUALIZATION, QLIKVIEW AND SCALABILITY

Cloud vendors offering the ability to host Windows Servers will make use of a virtualization technology to be able manage on demand instances of servers. QlikTech's approach to supporting virtualization is to focus on supporting Windows operating systems rather than the range of underlying virtualization technologies. So if the cloud vendor offers the supported versions of Windows in its offering then QlikView is supported to run on those.

Support notwithstanding, there are practicalities around scalability to be considered when using virtualization and thus cloud with QlikView and further consideration on what the current cloud vendors are able to offer with their own scalability.

QlikView is an in-memory software platform and so speed of access to RAM is a key factor in delivering good performance to users. Similarly, availability of CPU power to perform aggregations on the fly and produce content for users is also key. Virtualization technologies place some restrictions of speed of access to RAM compared to physical servers and can also limit the number of CPUs available to a single machine; this is reflected in the maximum size that cloud servers can be with regard to RAM and CPU. This can mean that not all QlikView deployments are suited to being virtualized and so it is important to size a deployment of QlikView when considering use of cloud.

Although a single machine may be limited, one of the significant advantages of cloud infrastructure is the ability to dynamically scale out onto multiple servers to accommodate demand. QlikTech's own internal sites make use of this capability to accommodate for peaks in demand around significant company events. This type of ease of scalability is unique to cloud deployments and very appealing when using QlikView within it.

QlikView & Cloud Vendors

QLIKVIEW ON AMAZON EC2

Summary

Amazon, through its EC2 service, provides Infrastructure as a Service (hosted, on demand virtual servers) as well as several other cloud bases services through its Amazon Web Service (AWS) relating to storage and applications. Details on all their services can be found on their web site (http://aws.amazon.com/products/) but the focus for a QlikView deployment is EC2.

The EC2 Cloud Compute offering allows customers to create virtual machines in the cloud on demand and pay for the length of time the machine runs. The instances start from one of a number of prebuilt images using Windows or Linux plus a number of other options such as preinstalled SQL Servers. Customer can also build their own images know as AMIs (Amazon Machine Images) to create images for immediate deployment.

Each instance when deployed is given an amount of resources and Amazon labels these from Micro to Quadruple Extra Large with differing amounts of RAM and CPU. The largest instance at time of writing can have 68GB of RAM and 8 CPU cores. (Amazon refers to "compute units" rather than cores but 8 Cores is what is visible to a Windows operating system.)

Customers can create and manage instances, their configuration and security using the browser EC2 Management Console. There are a huge range of features to support, manage and maintain a cloud environment on Amazon including dynamically adding new servers when needed, backup and storage.

For use with QlikView a customer would create a Windows instance with sufficient resources and install QlikView onto that Windows host and make it available through the EC2 firewall Although there is some terminology to learn, creation of instances only requires a web browser and no code or deep understanding of the underlying technologies is required.

Data and Storage

Amazon provides a number of mechanisms for data and storage.

The S3 storage cloud is akin to a virtual drive that you can add content to. It can be accessed through URLs, with code via an API or just mounted as a drive letter into a virtual machine. The storage is highly resilient and flexible.

The Dynamo DB, Map Reduce and RDS services are database mechanisms offered by AWS however these are not databases in the classic sense such as a SQL Server. The Map Reduce service is a Hadoop framework offering.

With AWS the data option that may get used depends on what data a customer may be reporting on and how it will be updated. They may not be used at all in most scenarios.

Security

EC2 does not offer an authentication service as such. EC2 does provide a range of facilities to secure the infrastructure however.

QLIKVIEW ON MICROSOFT AZURE

Microsoft offer through its Azure product a Platform as a Service. While they do offer a hosted Virtual Machine capability the core is based around the dynamic hosting of web applications and SQL databases. The intention is that the customer doesn't have to touch the Windows operating system that runs either the application or the database.

The Azure platforms' features lean more towards providing flexibility and scalability to applications rather than infrastructure but the features are rich and wide.

Similar to EC2 a customer can choose to place their applications to run on an instance of a given size, the largest at the time of writing being 16GB RAM with 8 cores. The charging model is different for Azure (compared to Amazon) with details provided on their web site.

What is distinctly different is how the instances are created; these are detailed below with their impacts on using QlikView within them.

Web Role

A web role is intended to host a web based application within IIS (Internet Information Services) on a Windows Server. To deploy to a web role a customer builds an Azure package containing the application, uploads it to Azure and then decides how much resource to give to that package. The amount of resources and number of instances can be flexed as needed and there are several features to support this.

While QlikView is accessed by users as a web application the back end is based on several Windows services and these are installed via an installer. To package QlikView into a web role the package must run the installer silently and then use the management API to license and configure the server. While this takes some knowledge of the Azure platform API it is possible to deploy QlikView in this way.

This style of deployment is well suited to deploying a configurable and packaged QlikView environment and so may appeal best to QlikView partners wanting to offer a similar offering to their customers. Once packaged a fully functional QlikView Server can be made available within 10 minutes and the same package could be used to deploy multiple servers with different configurations.

VM Role

Azure also offers the ability to host standard virtual servers in the cloud. However unlike the EC2 offering the Virtual Machine must be built on-premises before being uploaded to Azure. This means for each change the VM must be modified and uploaded. In addition the VM storage is not persistent, so if a VM is running and adding content to its local disk, this data would be wiped if the machine was ever restarted. Although a solution could be worked around this the web role is more suitable to QlikView.

Data and Storage

The Azure platform offers cloud based storage that in a similar way to EC2 can be written to via an API, attached as a drive letter to a server or accessed via a URL.

In addition the Azure platform has a SQL Azure component for hosting SQL databases in the cloud. One unique feature of this is the ability to automatically sync on-premise SQL databases with one in the cloud. Similarly a SQL Azure database could be used as a staging point to have data reload into QlikView.

An Azure deployed application is not persistent with storage and this means that if the box is redeployed it is done so from the package, so essentially it is like starting with a clean new server. For content that must persist using blobs of Azure storage is required and these need to be mounted as drives into the machines.

Security

As well as a range of infrastructure security features the Azure platform offers a number of features relating to authentication and security that can be used by a QlikView deployment. The Azure platform provides a number of features under its Access Control Service (ACS) which allows customers to integrate applications with standards-based identity providers, including enterprise directories such as Active Directory, and web identities such as Windows Live ID, Google, Yahoo!, and Facebook. There are a range of examples provided by Microsoft also.

When to Deploy QlikView in the Cloud

We offer some guidance for customers considering deploying QlikView in the cloud.

- **Follow the data.** If all the data users want to analyze is on-premise, the cloud may be less relevant to the QlikView deployment. Moving the data into the cloud and keeping it fresh may not be worth the effort. However, many QlikView OEM partners offer data services, either enhancing the customer's data or compiling and selling access to cloud-based data. If the data is in the cloud, it makes perfect sense for the QlikView app to also be in the cloud.

- **Make sure the pros outweigh the cons.** Customers may not want to deploy QlikView in the cloud if the following concerns overwhelm the potential benefits: data sensitivity (privacy, security), data integration and cloud orchestration costs, data model destandardization risk, unclear migration to on-premise BI, additional focus needed on vendor management, lack of integration among apps, and data protection laws which vary by country.

For apps in which sensitive data is being combined with cloud data, QlikView can run on-premise or in a private cloud. Apps fed by cloud data can run in the cloud. Customers get the best of both worlds: a flexible approach in which some Business Discovery apps are in the cloud and some are on premise.

End Notes

1. Source: Forrester Research, Sizing the cloud, April 21, 2011.

2. QlikView Security Overview Technology White Paper:
   http://www.qlikview.com/us/explore/resources/whitepapers/qlikview-security-overview

3. QlikView Community: http://community.qlikview.com/index.jspa