HCL VersionVault in Azure

Author: Rick Kissel
Date: 2020-08-13
Contact: DevOpsInfo@hcl.com

Contents

1 Summary

HCL VersionVault™ can run successfully in the Azure Cloud. Azure provides virtual machines (VMs) and other services (e.g., Azure Backup, Azure Files, and Azure Active Directory) that can be used to run HCL VersionVault servers and clients. Azure VMs support Windows and Linux OS versions that HCL VersionVault also supports so a complete HCL VersionVault installation using those OS versions can be setup in Azure.

This paper will discuss various options for configuring your set of Azure virtual machines so that HCL VersionVault will perform well. In general, all the HCL VersionVault recommendations for client and server sizing (for CPUs, memory, disk, network latency, etc.) also apply to Azure VMs. For local clients (dynamic or snapshot views) this will usually require that the clients and the servers they use run in the same Proximity Placement Group. For remote clients (automatic or web views) the clients can be separated from the servers they use in different Azure Regions or between Azure and on-premises networks. Azure documentation and tools should be consulted when setting up HCL VersionVault in Azure in order to meet those recommendations, determine costs, etc.

2 Introduction

This paper is intended for administrators of HCL VersionVault installations who would like to deploy HCL VersionVault to Azure. Familiarity with HCL VersionVault configuration and administration is assumed.

This paper starts with an overview of some Azure capabilities and terminology, followed by a section describing how those capabilities can be used for a HCL VersionVault deployment, and then a section with some sample deployment scenarios.

3 Overview of Some Azure Capabilities

This section summarizes some of the Azure capabilities and terminology that would be useful to HCL VersionVault administrators as they consider using Azure to support their HCL VersionVault installations. Later sections will discuss using these Azure capabilities for specific parts of a HCL VersionVault installation. The Azure Website provides complete information about Azure and its capabilities. Some useful documents for understanding Azure and its capabilities are Introduction to cloud computing and Microsoft Cloud Adoption Framework for Azure.

3.1 Azure Virtual Machines

This is an annotated outline of links that follows the information in the Azure Virtual Machines document and provides links to other information in the Azure Documentation. Note, there is documentation for Linux Virtual Machines and Windows Virtual Machines that are very similar and many of the links below point into the Linux user guide for their information.

- Virtual Machines
  - Example Azure infrastructure walkthrough for Linux VMs provides an example that references the following VM concepts.
  - Regions and Availability Zones

    A Region (geographical regions) is usually called a *location* when you create a VM. The location specifies where the virtual hard disks are stored.

    An Availability Zone (AZ) is a high-availability offering at a unique physical location within a recommended region (there are at least 3 AZs per recommended region). Alternate (other) regions are not designed to support AZs

    Use a Proximity Placement Group to get VMs as close as possible, achieving the lowest possible latency.
  - VM Images

An image is a copy of either a full VM (including any attached data disks) or just the OS disk, depending on how it is created. There are generalized and specialized images that have different trade-offs when they are used.

- You can find Linux and Windows images in Azure Marketplace Images.
- For Linux, there are Endorsed Linux distributions and Information for Non-endorsed Linux Distributions.
- For Windows, more information on the supported guest operating systems, roles, and features can be found in Microsoft server software support for Microsoft Azure virtual machines.
- Create a managed image of a virtual machine or VHD.
- You can manage your images with a Shared Image Gallery.

- VM Disks

Only the Hyper-V fixed virtual hard disk (VHD) format is supported for VM images and disks.

- VM disks use page blobs for the *operating system disk*, the *temporary disk*, and *data disks* as described in Manage Azure disks with the Azure CLI.
- By default, VM disks use Azure Managed Disks. If you have unmanaged disks, see: Convert a Linux virtual machine from unmanaged disks to managed disks, and for Windows: Migrate Azure VMs to Managed Disks in Azure.

  You can also create the VM disks as unmanaged disks, which is not recommended. If you create a VM with unmanaged disks, make sure that you attach disks from storage accounts residing in the same region as your VM to ensure close proximity and minimize network latency. See Back up Azure unmanaged VM disks with incremental snapshots for more information about using unmanaged disks.

  The rest of this paper will assume you are using Azure Managed Disks.

- Azure Blob storage

VM disks use blob storage as their underlying storage mechanism.

- Blob storage overview
- Blob storage introduction
- Blob types

- Azure Files

Provides shareable SMB-based file storage that can be used by your VM and shared with on-premises deployments, as well.

- Azure NetApp Files

Supports NFS and SMBv3 volumes. See Create an SMB volume for Azure NetApp Files and Create an NFS volume for Azure NetApp Files.

- Networking and Security

Azure provides many features for networking and security, including the following:

- Azure Virtual Network

VNet is the fundamental building block for your private network in Azure. It allows you to communicate with your VMs, the internet, and on-premises networks in a scalable, available, and isolated way.

- Azure Active Directory

Azure AD is Microsoft's cloud-based identity and access management service. It can help you control access to external resources (e.g., MS Office 365, Azure portal) and internal resources on your corporate network.

- Management and Governance

Azure provides many features for managing your deployment, including the following:

- Azure Monitor

  A comprehensive solution for collecting, analyzing, and acting on information from your cloud and on-premises deployments.

- Azure Resource Manager

  ARM provides a management layer that enables you to create, update, and delete resources in your Azure account, including your VMs. You can use features, like access control, locks, and tags, to secure and organize your resources after deployment.

3.2 Windows Virtual Desktop

Windows Virtual Desktop is a desktop and app virtualization service that runs on the cloud. There is no similar offering for a Linux desktop.

4 Plan Your HCL VersionVault Deployment in Azure

Given some of the Azure services and products discussed above, we can now talk about how you might use those capabilities to deploy HCL VersionVault in Azure.

Determining key aspects of your HCL VersionVault deployment will provide information that will be useful in making decisions about how Azure services can be used. The Deploy HCL VersionVault document provides more information about things you should consider including a description of the requirements for deployment.

4.1 HCL VersionVault Servers (VOBs/Views/Registry)

- What hardware and OS platforms do you use for VOB and view servers?
  - Linux (what distros and hardware)?
  - Windows (what versions)?
  - Others?
- Where will the servers run?
  - Azure?
  - On-premises?
  - Azure and on-premises?
  - Do you use HCL VersionVault™ MultiSite?
- Where will the server storage reside?
  - local server disk?
  - on a SAN-type device?
  - on a NAS-type device?
- How do you access VOB and view storage from servers?
  - Local file system?

    - NFS?
    - SMB (CIFS)?
    - Both? To the same storage?

4.2 HCL VersionVault Clients

- What hardware and OS platforms do your clients use?
    - Linux (what distros and hardware)?
    - Windows (what versions)?
    - Others?
- What view platform(s) do your clients use?
    - Unix/Linux only views?
    - Windows only views?
    - Both?
- What view type(s) do your clients use?
    - Local view types (dynamic or snapshot)?
    - Remote view types (web or automatic)?
    - MVFS-based views (dynamic or automatic), which require in-kernel installation?
- How do your local view types access VOB and view storage from clients?
    - NFS?
    - SMB (CIFS)?
    - Both? To the same storage?
- Where will the servers run and where will their storage reside?
    - Azure?
    - On-premises?
    - Azure and on-premises?
- Do you run large clients as “build servers” (multi-user, multi-view clients)?

5 Azure Considerations

Microsoft provides a wealth of documentation for Azure. A good starting place would be Cloud Adoption Framework for Azure, which describes and links to topics like:

- What is the Cloud Adoption Framework
- Get started with the Cloud Adoption Framework
- Understand the fundamental concepts around cloud adoption
- How Azure works
- Migrate existing workloads to the cloud
- Azure setup guide
- Azure foundational concepts

5.1 Azure VMs and Images

Azure VMs are created from images. For example, see Create a Linux virtual machine with the Azure CLI. Azure has images defined for Windows that run on 32-bit or 64-bit x86 and for Linux that run on 64bit x86. There are images that support Windows and Linux OS versions that HCL VersionVault also supports, e.g., Windows 10 and RHEL 8.2 on x86_64, so a complete HCL VersionVault installation using those OS versions can be setup in Azure. There are also images that support OS versions that HCL VersionVault does not support, e.g., Fedora Linux, Kali Linux, Debian, or OS versions on arm64, so those images cannot be used to create HCL VersionVault client or server VMs. You can check the requirements for HCL VersionVault to see what hardware and OS versions are supported by HCL VersionVault. Once deployed, a VM can be updated to newer OS versions, or even installed with different OS versions that HCL VersionVault supports. The Shared Image Gallery may be a good option for managing a collection of your images.

Since customers can create their own image from a running VM that has been customized, they can create an image containing a complete HCL VersionVault client or server and then use that image to create subsequent VMs that have HCL VersionVault already installed and configured. In general, you won't want to duplicate a complete HCL VersionVault server, e.g., having multiple registry servers with the same configuration will cause problems. However, you could have a "base" configuration that didn't include particular server details to provide a good starting point for configuring each server VM.

Azure lets you create generalized and specialized images. Generalizing is a process that removes machine and user specific information from the VM. For Windows, the Sysprep tool is used. For Linux, you can use waagent -deprovision or -deprovision+user parameters. Specialized VMs have not been through a process to remove machine specific information and accounts.

To create and manage Azure virtual machines (VMs) in a consistent manner at scale, some form of automation is typically used. A number of tools you can use in Azure, like Ansibile, Chef, Puppet, Cloud-init, Packer, ARM templates, etc., are described in Use infrastructure automation tools with virtual machines in Azure.

One way to implement *infrastructure as code* for your Azure solutions is to use Azure Resource Manager templates (ARM templates). The template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax, which lets you state what you intend to deploy without having to write the sequence of programming commands to create it. In the template, you specify the resources to deploy and the properties for those resources.

You could also follow How to use Packer to create Linux virtual machine images in Azure as another way to create a VM image.

5.2 VMs versus Windows Virtual Desktops

A Windows Virtual Desktop can be used for easy Windows desktop client management, and it might provide a lower cost than a VM. This option is not available for a Linux desktop. You can customize a master image for use with Windows Virtual Desktop, including installing your own applications (e.g., HCL VersionVault).

5.3 VM sizes

Azure provides many sizes for virtual machines that can be used for VMs. The minimum recommended hardware requirements for HCL VersionVault clients and servers can be found in HCL VersionVault Platform Requirements and Deploy HCL VersionVault.

The various Azure general purpose VM sizes are normally suitable for HCL VersionVault. Azure memory optimized VM sizes may be considered for heavily used VOB and view servers because the HCL VersionVault database used by these servers can benefit from more (or faster) memory. Burstable performance VM sizes may be a good choice for clients with workloads that vary and may be more "bursty". Azure dedicated hosts may be of interest for servers for which a site would like more control over maintenance events or where hardware isolation is desired.

Azure Monitor provides information that you can use to evaluate your VMs to help you decide on the most cost effective VM sizes, storage, networking, etc., to use for your environment. Note, you can resize a virtual machine using Azure CLI. Thus, as your needs evolve (or you've made a mistake), you can change the VM size to a more appropriate one without losing your VM.

5.4 VOB and View Storage

Azure Managed Disks are the "local disk" persistent storage for VMs, either Windows or Linux, and are appropriate for HCL VersionVault VOB and view storage. As with on-premises HCL VersionVault, this means that each HCL VersionVault server needs to export its storage (e.g., through SMB and/or NFS) so that remote clients can access it.

Azure NetApp Files could be used in a mixed Linux and Windows environment, or in a homogeneous environment, since they can be accessed through both NFS and SMB. As with on-premises HCL VersionVault using remote storage (like a NetApp filer), the network latency between the remote storage (the Azure NetApp Files in this case), and the clients and servers using the remote storage must be suitable for the HCL VersionVault application as described below.

5.5 Communication between Azure and on-premises networks

Azure networking services provides an overview that describes the various connection options between Azure and on-premises networks. In particular, on-premises networks might want to use a VPN Gateway or ExpressRoute to connect to the Azure cloud. Note, the local view type clients and all servers for HCL VersionVault need to be protected from outside (public) access by a firewall or other similar technology. Remote view type clients could be outside of a firewall. Application protection services provides an overview of the Azure capabilities for managing access across networks.

For HCL VersionVault, the most important network parameter for dynamic view performance is the latency between clients and servers and also between some servers like view servers and VOB servers. Thus, in general, splitting clients and servers between on-premises machines and Azure VMs will not provide acceptable performance for dynamic views.

5.6 Network latency: intra-AZ, intra-region, inter-Region

Network latency between clients and servers is extremely important for acceptable performance of HCL VersionVault local view types (dynamic and snapshot). Network latency between servers (view, VOB, registry, etc.) is also extremely important for acceptable performance of all HCL VersionVault view types (dynamic, snapshot, web, and automatic). Network latency between remote clients (with view types web or automatic) and the servers is much less important.

Placing VMs in a single region limits the physical distance between them. Placing VMs in a single AZ will usually put them in the same Azure datacenter. However, as an Azure region grows, a single AZ may expand to cover multiple datacenters. This may increase the network latency between VMs in a single AZ and cause HCL VersionVault performance to be unacceptable.

To get get the minimum latency between VMs, you should deploy them within a Proximity Placement Group.

Azure provides no network latency guarantees, but experience shows that:

- For VMs in a Proximity Placement Group, network latency is suitable for any HCL VersionVault communication.
- For VMs in the same Availability Zone, network latency should be suitable for any HCL VersionVault communication, but should be measured if performance problems are seen.
- For VMs in the same region, network latency may or may not be suitable for HCL VersionVault communication (it may depend on the region) and should be measured before using.

- For VMs in different regions, or for Azure to customer on-premises connections, network latency is probably too high for most HCL VersionVault communication, other than for remote clients using automatic or web views or for HCL VersionVault MultiSite.

5.7 Security, access control, and Active Directory

General OS-level access control across VMs is described in Security best practices for IaaS workloads in Azure, which covers both Linux and Windows and recommends ways to manage users and their access. Azure provides Azure AD as their main authentication and authorization service. Compare self-managed Active Directory Domain Services, Azure Active Directory, and managed Azure Active Directory Domain Services compares Azure AD solutions. Azure AD can also be integrated with your on-premises AD if necessary. Log in to a Linux virtual machine in Azure using Azure Active Directory authentication provides information specific to using Azure AD for Linux.

HCL VersionVault can use AD in Azure with the same considerations that are needed for AD use on-premises. For instance, in order for Windows clients to use VOBs/views on Unix/Linux, there must be a way to map the Windows user's identity to a Unix/Linux user identity. AD is often used to provide identities for both Windows and Unix/Linux (through its LDAP interface) to make it easier to keep these identities synchronized. See Plan your deployment of HCL VersionVault for details on deploying HCL VersionVault.

Of course, the customer can set up their own infrastructure (e.g., Microsoft AD or NIS) on their own VMs and manage OS users and groups themselves, just as they would in an on-premises network.

On Linux, individual users could be created and managed on each VM, e.g., by using the Linux useradd command. This might be suitable for a small HCL VersionVault setup and would avoid using a centralized directory service.

5.7.1 Network and firewall considerations

For HCL VersionVault within Azure you need to make sure that the Network Security Groups used by your VMs allow *nfs* (port 2049), *sunrpc* (port 111), and *albd* (port 371) access through both TCP and UDP. You also need to make sure that any firewall software running on each VM (e.g., iptables on Linux) also allows those protocols. Given that some protocols, including HCL VersionVault view and VOB server RPCs, also use dynamically assigned non-privileged ports, you will need to open all non-privileged ports in the security groups and firewalls that control network access between your VMs. See Issues when using Windows Firewall for more Windows-specific information.

On Linux with dynamic view clients, you will also need NFS and automount running to allow the clients to access remote VOB and view storage through NFS. You will also have to set up NFS exports on the view and VOB servers to allow NFS access from the clients.

On Windows with dynamic view clients, you will need to allow sharing of VOB and view storage over SMB/CIFS (ports 139 and/or 445) between the clients and the servers.

5.8 HCL VersionVault MultiSite

HCL VersionVault MultiSite can be used in Azure. Customers might need to add (or extend) HCL VersionVault MultiSite to servers running on Azure VMs to provide dynamic view access to the VOB replicas that HCL VersionVault MultiSite creates, just as they would to provide that access between widely separated (high latency) on-premises servers. This is because the Azure inter-region network latency or the network latency between Azure and an on-premises network is high enough that it would affect dynamic views when accessing the "remote" servers. HCL VersionVault MultiSite is designed to alleviate this problem by allowing the dynamic view clients and the VOB replicas they access to be located in Azure such that they have a low latency connection (e.g., in the same Availability Zone).

For Linux and Unix systems, HCL VersionVault MultiSite can be configured to operate through a firewall using the shipping server. There is also information for configuring MultSite shipping server to work within a static port range on both Unix/Linux and Windows that may be helpful when setting up a firewall configuration. This may be needed for configuring HCL VersionVault MultiSite to operate between

on-premises and Azure HCL VersionVault installations.

5.9 Backups

The HCL VersionVault administration documentation provides a section on Backing up critical HCL VersionVault data that provides the information necessary for backing up HCL VersionVault VOBs and views. The use of managed disk Linux Snapshots or Windows Snapshots of the storage on particular servers could be used as part of your backup implementation. You may also be able to use an incremental snapshot. For VOB data, using HCL VersionVault MultiSite to replicate to servers in a different Azure Region could be part of your backup solution. This could be used to avoid a single point of failure in the Azure infrastructure because Azure regions are relatively independent.

5.10 Resilience

Azure provides lots of information on resilience considerations including Azure Resiliency and Overview of the reliability pillar.

5.11 Monitoring

Azure Monitor provides a number of services that can be used to gauge the health of the various servers and clients running in Azure. They can also provide notifications if various configured thresholds are exceeded.

5.12 Performance

A few performance considerations have been mentioned in previous sections, so those should be kept in mind when configuring your Azure environment. There are many options in Azure for configuring the the number of cpus, memory size, network bandwidth, dedicated vs. shared VMs, etc. (see sizes for virtual machines).

In general, HCL VersionVault performance in Azure is affected by the same factors as HCL VersionVault performance in an on-premises installation. This blog (part 1 and part 2) provides some principles and techniques for HCL VersionVault performance analysis and tuning. There is also HCL VersionVault documentation for client performance tuning and VOB server performance tuning. Some earlier recommendations for HCL VersionVault in a virtual environment can be found in General Virtualization Considerations (pt 1) and General Virtualization Considerations (pt 2).

5.13 Servers, clients, or both?

The key factor to consider for a HCL VersionVault deployment is the network latency between the various pieces of the deployment (clients, servers, VOB and view storage, etc.). For this purpose, the network latency within an Azure Availability Zone for machines in the same Azure Virtual Network should be acceptable for any HCL VersionVault communications among clients, view servers, VOB servers, registry servers, etc. The network latency between AZs in the same region may or may not be be acceptable for any HCL VersionVault communications; it should be measured to determine its acceptability. The network latency between Azure regions and between Azure and the customer on-premises network will generally be too long to allow anything other than HCL VersionVault remote client access.

Configuring replicated VOBs using HCL VersionVault MultiSite could be used allow VOB servers to be “close enough” to clients and other servers to allow dynamic view usage of the multi-sited VOBs by clients in Azure as well as clients in the on-premises network. If your on-premises network has a firewall, you may be able to use the information in Store-and-forward through a firewall (Linux and the UNIX system only) to allow HCL VersionVault MultiSite to operate through the firewall.

6 Sample Usage Scenarios

What follows are some possible customer scenarios for using Azure for HCL VersionVault. These scenarios assume the customer has suitable access to clients and servers running in Azure from their on-premises network, typically Remote Desktop for Windows clients/server and VNC and/or SSH for Linux clients/servers. There are major differences between HCL VersionVault local view types (dynamic and snapshot) and remote view types (web and automatic), so the scenarios are divided into those two major categories.

6.1 Scenario 1: Local view types (dynamic and snapshot)

The customer has a current on-premises HCL VersionVault installation using dynamic views. They would like to move a subset, or all, of their clients and servers to Azure. Due to the expected network latency between the customer's on-premises network and Azure, this scenario only makes sense if the clients and servers to be moved are "isolated" from the clients and servers that remain in the on-premises network. That is, the clients that are moved should need access only to the VOB, view, and registry servers that are also moved, and the clients and servers that remain on-premises need access only to those clients and servers that remain on-premises. Note, this means that any moved registry servers would support different regions than the on-premises registry servers. Otherwise, if the clients that are moved also need access to on-premises servers, the performance of the clients accessing the on-premises views and VOBs will be unacceptable due to the high network latency between Azure and the on-premises network. The isolated subset of clients and servers should be moved to a single Availability Zone in Azure to ensure that the network latency between them is acceptably low.

Figure 1 shows a very simple on-premises HCL VersionVault configuration where the dynamic view client and its local storage are on one machine and all the servers and their local storage are on another machine. The communication between the client and the servers is low latency (over a LAN). The dotted ovals show how this configuration might appear in Azure with a VM and its associated disk for the client and another VM and its associated disk for the servers. These two VMs would be in the same AZ so that the communication between them would be low latency to allow the dynamic view client to perform well.

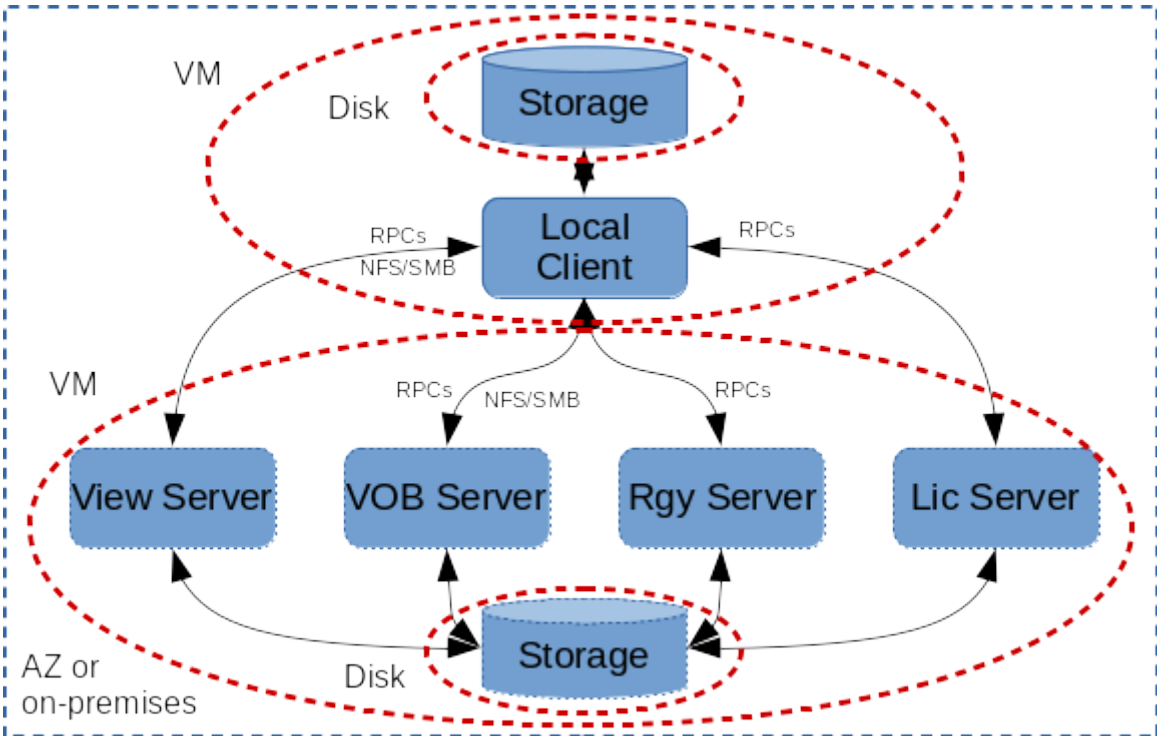

Figure 1: Local client on one machine and servers on another machine

6.2 Scenario 2: Remote view types (automatic and web)

The customer on-premises HCL VersionVault installation uses remote view types, i.e., automatic views or web views. In this case, clients could be moved to Azure and they could continue to access servers in the on-premises network because the network latency would not be a problem for remote view types.

Alternatively, servers could be moved to Azure with the clients continuing to run in the on-premises network. In this case, the CCRC server, as well as all the VOB and registry servers it accesses, would need to be moved to Azure in the same AZ. This is because the CCRC server needs low latency network access to the VOBs and registry servers that it uses.

Finally, both the clients and servers could be moved to Azure and the clients could be in different AZs, or even different Azure regions, from the servers. Again, the servers would need to be in the same AZ, as described previously.

Figure 2 shows a very simple on-premises HCL VersionVault configuration where the remote view client and its local storage are on one machine, the CCRC WAN Server is on another machine across a high latency connection (WAN) from the client, and all the other servers and their local storage are on another machine with low latency connections (LAN) between them and to the CCRC WAN Server. The dotted ovals show how this configuration might appear in Azure with a VM and its associated disk for the client, another VM and its associated disk for the CCRC WAN Server, and another VM and its associated disk for the other servers. The CCRC WAN Server VM and the other servers VM would be in the same AZ so that the communication between them would be low latency. The client VM could be “anywhere”, e.g., in another Azure region or on the on-premises network, since it is not network latency sensitive.

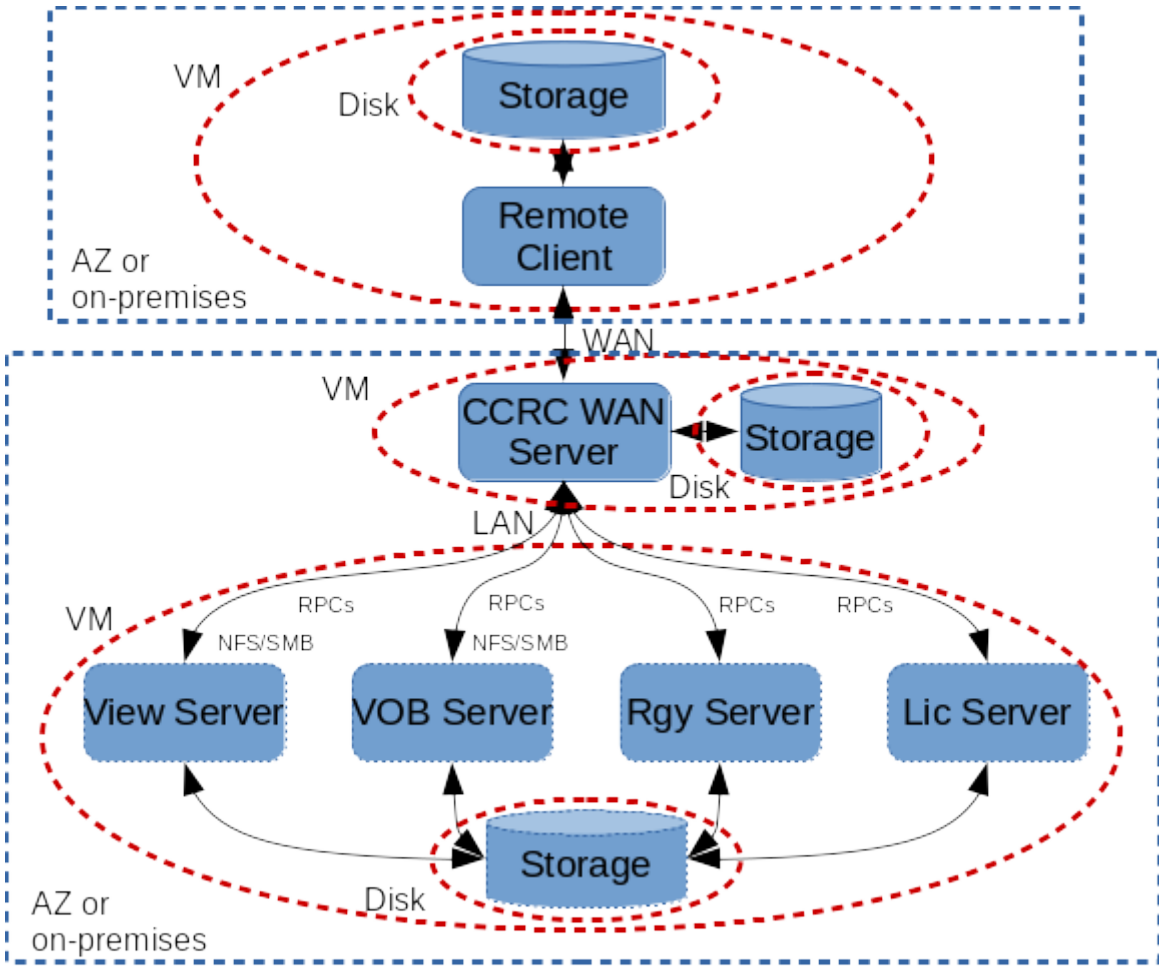

Figure 2: Remote client on one machine and servers on other machines

6.3 Scenario 3: Using HCL VersionVault MultiSite between on-premises and Azure

The customer has an existing on-premises HCL VersionVault installation using local view types and wants to add clients and servers in Azure that are synchronized by HCL VersionVault MultiSite. If there are firewall(s) between your on-premises network and Azure (e.g., on your on-premises network and/or on your Azure VNet), you will need the information in Store-and-forward through a firewall (Linux and the UNIX system only) to allow HCL VersionVault MultiSite to operate through the firewall(s).

Figure 3 shows a local view type setup on-premises and a local view type setup in Azure where the VOBs are synchronized using HCL VersionVault MultiSite (see Administering HCL VersionVault MultiSite for details).

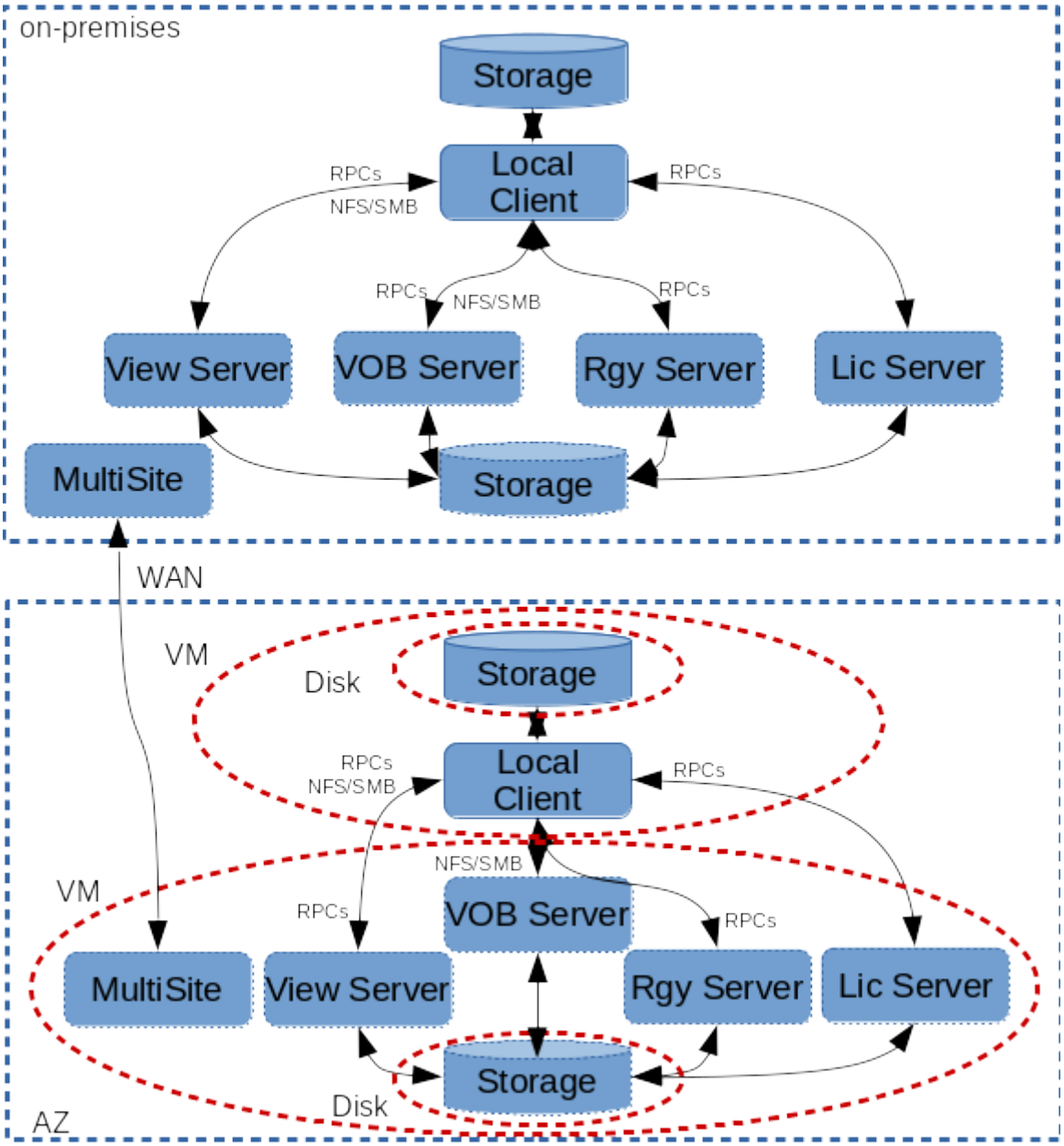

Figure 3: HCL VersionVault MultiSite between on-premises and Azure

6.4 Scenarios common between local and remote view types

In a mixed environment, the VOB and registry servers might be used both by CCRC servers for remote view type clients and by local view type clients. That is, if the VOB and registry servers are shared between local and remote view type clients, then the setup becomes more complicated. The CCRC server has to be “close” to its VOBs and registry servers and the local view clients would also have to be “close” to the same VOBs and registry servers. That is, all of the CCRC servers, local view type clients, and their shared VOBs and registry servers need to be all in the same Azure AZ or all in the on-premises network.

For scenarios where network latency is “too high”“, e.g., a hybrid cloud between on-premises and Azure or between VMs in different Azure regions, HCL VersionVault solutions for high latency networking can be used. These include remote clients using web or automatic views and HCL VersionVault MultiSite.

7 References And More Information

Azure Website - The main Azure website from which everything Azure related can be found.

Azure Documentation - The Azure documentation website for user guides, developer guides, etc.

Deploy HCL VersionVault - Information about planning, installing, and configuring a HCL VersionVault deployment, including requirements to identify system requirements, prerequisite tasks, and other information.

General Virtualization Considerations (pt 1) - Some general things to consider when virtualizing HCL VersionVault (or any application).

General Virtualization Considerations (pt 2) - Some (relatively old) performance measurements for HCL VersionVault running in a VMWare environment.

HCL VersionVault Platform Requirements - The IBM page where you can search for the product and then filter results as appropriate to find out what OS’s and platforms HCL VersionVault supports as well as “hardware” (real or virtual) requirements.

Knowledge Collection: Available White Papers for the VersionVault Family of Products - Provides useful information on various aspects of HCL VersionVault configuration and usage, e.g., VersionVault and Samba: A Supported Configuration.

Introduction to cloud computing - Azure information on cloud computing in general.