HCL VersionVault in GCP

Author: Rick Kissel
Date: 2022-11-21
Contact: DevOpsInfo@hcl.com

Contents

1 Summary

HCL VersionVault™ can run successfully in the Google Cloud Platform (GCP). GCP provides virtual machines (VM instances) through its Compute Engine service and also other services (e.g., Backup and Disaster Recovery, Filestore, and Managed Microsoft Active Directory) that can be used to run HCL VersionVault servers and clients. GCP VM instances 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 GCP.

This paper will discuss various options for configuring your set of GCP 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 GCP VM instances. For local clients (dynamic or snapshot views) this will usually require that the clients and the servers they use run in the same Zone (ideally, in the same Cluster). For remote clients (automatic or web views) the clients can be separated from the servers they use in different GCP Regions or between GCP and on-premises networks. GCP documentation and tools should be consulted when setting up HCL VersionVault in GCP 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 GCP. Familiarity with HCL VersionVault configuration and administration is assumed.

This paper starts with an overview of some GCP 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 GCP Capabilities

This section summarizes some of the GCP capabilities and terminology that would be useful to HCL VersionVault administrators as they consider using GCP to support their HCL VersionVault installations. Later sections will discuss using these GCP capabilities for specific parts of a HCL VersionVault installation. The GCP Website provides complete information about GCP and its capabilities. Some more specific information can be found in Google Cloud overview, Get Started with Google Cloud, and GCP Products Cheatsheet. What is Cloud Computing is a general introduction to cloud computing concepts. Google Cloud Architecture Framework has recommendations and best practices for designing and operating a cloud infrastructure.

3.1 GCP Compute Engine Service

This is an annotated outline of links that can be found by starting with the Compute Engine Documentation guides as well as links to other information in the general GCP Documentation.

- Machine Families

  The *machine family* determines the hardware characteristics for your VM instance, which include compute, memory, and storage capabilities.

- Regions and Zones

  Compute Engine resources are hosted in multiple locations worldwide. These locations are composed of regions and zones. A region is a specific geographical location where you can host your resources. Regions have three or more zones.

  Resources that live in a zone, such as VM instances, are referred to as zonal resources. Other resources, like static external IP addresses, are regional. Regional resources can be used by any resource in that region, regardless of zone, while zonal resources can only be used by other resources in the same zone.

- Zones and Clusters

Compute Engine implements a layer of abstraction between zones and the physical clusters where the zones are hosted. A cluster represents a distinct physical infrastructure that is housed in a data center. The Zone Virtualization document describes this in more detail. VM instances in the same cluster should have the lowest network latency between them.

- VM Instances

  An *instance* is a virtual machine (VM) hosted on Google's infrastructure.

- VM Instance Templates

  An *instance template* is a resource that you can use to create virtual machine instances (create individual VMs) and managed instance groups (create a MIG).

  Instance templates define the machine type, boot disk image or container image, labels, startup script, and other instance properties. Instance templates are a convenient way to save a VM instance's configuration so you can use it later to create VM instances or groups of VM instances.

  - You can find information about Linux and Windows public and custom images that can be used to create and start a VM instance in VM Images.

    - Public images are provided and maintained by Google, open source communities, and third-party vendors. By default, all Google Cloud projects have access to these images and can use them to create instances.
    - Custom images are available only to your Cloud project. You can create a custom image from boot disks and other images. Then, use the custom image to create an instance.

- VM Storage Options

  Compute Engine offers several types of storage options for your instances. Each of the following storage options has unique price and performance characteristics:

  - Zonal persistent disk: Efficient, reliable block storage.
  - Regional persistent disk: Regional block storage replicated in two zones.
  - Local SSD: High performance, transient, local block storage.
  - Cloud Storage buckets: Affordable object storage.
  - Filestore: High performance file storage for Google Cloud users.

  If you are not sure which option to use, the most common solution is to add a persistent disk to your instance.

- VM Data Protection Options

  Compute Engine provides various options for backing up and replicating your persistent disk data.

- VM Networking Overview

  This provides an overview of networking functions for VM instances including Virtual Private Cloud networking (VPC network overview).

- VM Access Control Overview

  Manage user access to Compute Engine resources.

- Cloud and VM Monitoring

  Compute Engine provides a number of logs and other monitoring tools to track VM activity and resources. For instance, audit logs, usage reports, and license reporting. An example of monitoring an Apache Web Server on a VM instance is described in Monitor a Compute Engine VM

4 Plan Your HCL VersionVault Deployment in GCP

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

Determining key aspects of your HCL VersionVault deployment will provide information that will be useful in making decisions about how GCP 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?
    - GCP?
    - On-premises?
    - GCP 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?
    - GCP?
    - On-premises?
    - GCP and on-premises?
- Do you run large clients as “build servers” (multi-user, multi-view clients)?

5 GCP Considerations

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

- System Design
- Operational Excellence
- Security, Privacy, and Compliance
- Reliability
- Cost Optimization
- Performance Optimization

5.1 GCP VM Instances and Templates

GCP VM Instances are created from VM Instance Templates or you can Create Custom Images. For example, see Create and Start a VM Instance. GCP has public images defined for Windows and Linux that have OS versions that HCL VersionVault also supports, e.g., Windows Server 2019 and SLES 15.4 on x86_64, so a complete HCL VersionVault installation using those OS versions can be setup in GCP. There are also images that support OS versions that HCL VersionVault does not support, e.g., Fedora Linux, Debian, or OS versions on arm64, so those images cannot be used to create HCL VersionVault client or server VM instances. 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. Manage Access to Custom Images provides information about 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 VM instances 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.

GCP normally creates a VM instance from a VM template as a copy of the template with the exception of the instance name and zone. You can also create a VM instance from an instance template with overrides. This lets you override any property you want by passing a new value for the attribute when creating the instance. For instance, using the command:

```
gcloud compute instances create <instance name> --source-instance-template <template name>
```

you can override any property that can be specified with the gcloud command.

GCP allows you to specify metadata to the VM instance when you create it. If an instance has the Compute Engine tools installed (which the official images do) then some metadata keys have a special meaning. In

particular the `startup-script` key lets you specify a script to run when the VM instance starts up. See Running startup scripts for more information.

GCP also allows you to create VM instances by importing images from AWS, importing virtual appliances (in an OVF package or an OVA single file), and importing vitual disks (including VMDK and VHD disk formats). You can also migrate your VMware-based applications to the GCP CMware Engine.

5.2 VM Machine Families

GCP provides machine families and CPU platforms that can be used for VM instances. In general, HCL VersionVault clients and servers don't have particularly stringent requirements for the hardware on which they run (see HCL VersionVault Platform Requirements and Deploy HCL VersionVault).

The various GCP general purpose machine families are normally suitable for HCL VersionVault. Some heavily used VOB and view servers may benefit from the memory optimized machine families because the HCL VersionVault database used by these servers can benefit from more (or faster) memory. There are also sole-tenant nodes or a bare metal solution that could be appropriate for very heavily used VOB or view servers.

GCP provides various audit logs, usage reports, machine type recommendations, and other services that you can use to evaluate your VM instances to help you decide on the most cost effective machine families to use for your environment. Note, you can change the machine type for a VM instance by stopping it, changing the machine type, and restarting it. Thus, as your needs evolve (or you've made a mistake), you can change the machine type to a more appropriate one without losing your VM instance.

5.3 VOB and View Storage

GCP provides a number of storage options for VM instances. Zonal persistent disks or regional persistent disks are the "local disk" persistent storage for VM instances, 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.

GCP also provides filestore instances that are fully managed NFS file servers. They could be used somewhat like NAS storage in a homogeneous environment for VOB and view storage. For use in a mixed Linux and Windows environment the Windows servers would need to support NFS (see Mounting a file share on a Windows VM instance for more information).

5.4 Communication between GCP and on-premises networks

Under the GCP Architecture Framework: System design the section on design your network infrastructure covers considerations for setting up hybrid cloud networking between your on-premises networks and GCP. In particular, on-premises networks might want to use a dedicated interconnect or partner interconnect to connect to the GCP VPC. 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. Secure your network provides an overview of the GCP 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 GCP VM instances will not provide acceptable performance for dynamic views.

5.5 Network latency: intra-zone, 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 VM instances in a single region limits the physical distance between them. Placing VM instances in a single zone will usually put them in the same GCP datacenter. However, as an GCP region grows, a single zone may expand to cover multiple datacenters. This may increase the network latency between VM instances in a single zone and cause HCL VersionVault performance to be unacceptable.

To get get the minimum latency between VM instances, you should deploy them with compact placement policy.

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

- For VM instances using a compact placement policy, network latency is suitable for any HCL VersionVault communication.
- For VM instances in the same zone, network latency should be suitable for any HCL VersionVault communication, but should be measured if performance problems are seen.
- For VM instances 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 VM instances in different regions, or for GCP 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.

5.6 Security, access control, and Active Directory

GCP uses key-based SSH to connect to Linux VM instances. You can also enable SSH for Windows VM instances. You can manage access to VM instances through metadata or OS Login. OS Login is available only for Linux VM instances. OS Login uses GCP Identity and Access Management (IAM) for authentication rather than using SSH keys.

By default, GCP uses custom project and/or instance metadata to configure SSH keys and to manage SSH access. All Windows VM instances use metadata to manage SSH keys, while Linux VM instances can use metadata keys or OS Login.

GCP provides a Managed Microsoft AD service as one way to manage Active Directory for your VM instances. There is a Best practices for running AD on Google Cloud document that describes various ways of designing your AD infrastructure. GCP also provides GCP directory sync for synchronizing your GCP IAM users and groups with AD, e.g., if you are using OS Login on Linux VM instances and also need those identities in your AD server.

HCL VersionVault can use AD in GCP 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 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 VM instances and manage OS users and groups themselves, just as they would in an on-premises network. If you want to integrate an AD in GCP with your on-premises AD see AD in a hybrid environment.

On Linux, individual users could be created and managed on each VM instance, 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.

5.6.1 Network and firewall considerations

For HCL VersionVault within GCP you should consult the Secure your network architectural framework document. In particular, you need to make sure that the VPC firewall rules used by your VM instances 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 instance (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 VM instances.

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.

5.7 HCL VersionVault MultiSite

HCL VersionVault MultiSite can be used in GCP. Customers might need to add (or extend) HCL VersionVault MultiSite to servers running on GCP VM instances 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 GCP inter-region network latency or the network latency between GCP 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 GCP such that they have a low latency connection (e.g., in the same zone using the compact placement policy).

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 GCP HCL VersionVault installations.

5.8 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 Backup and disaster recovery document provides information about these GCP services. The use of persistent disk snapshots of the storage on particular servers could be used as part of your backup implementation. For VOB data, using HCL VersionVault MultiSite to replicate to servers in a different GCP region could be part of your backup solution. This could be used to avoid a single point of failure in the GCP infrastructure because GCP regions are relatively independent.

5.9 Resilience

GCP provides lots of information on resilience considerations including Resilient workloads with regional persistent disks, Patterns for scalable and resilient apps, and Architecting disaster recovery for cloud infrastructure outages.

5.10 Monitoring

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

5.11 Performance

A few performance considerations have been mentioned in previous sections, so those should be kept in mind when configuring your GCP environment. There are many options in GCP for configuring the the number of cpus, memory size, network bandwidth, dedicated vs. shared VM instances, etc. (see machine families for more details).

In general, HCL VersionVault performance in GCP is affected by the same factors as HCL VersionVault performance in an on-premises installation. This HCL VersionVault documentation for client performance tuning and VOB server performance tuning provides some general performance information.

5.12 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 a GCP zone using the compact placement policy for machines in the same VPC should be acceptable for any HCL VersionVault communications among clients, view servers, VOB servers, registry servers, etc. The network latency within a zone without the compact placement policy may also be acceptable but should be measured and evaluated. The network latency between zones in the same region may or may not be be acceptable for any HCL VersionVault communications. The network latency should be measured to determine if it is acceptable. The network latency between GCP regions and between GCP 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 to 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 GCP 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 GCP for HCL VersionVault. These scenarios assume the customer has suitable access to clients and servers running in GCP 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 GCP. Due to the expected network latency between the customer's on-premises network and GCP, 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 GCP and the on-premises network. The isolated subset of clients and servers should be moved to a single zone (or zone using the compact placement policy if necessary) in GCP 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 GCP with a VM instance and its associated disk for the client and another VM instance and its associated disk for the servers. These two VM instances would be in the same zone so that the communication between them would be low latency to allow the dynamic view client to perform well.

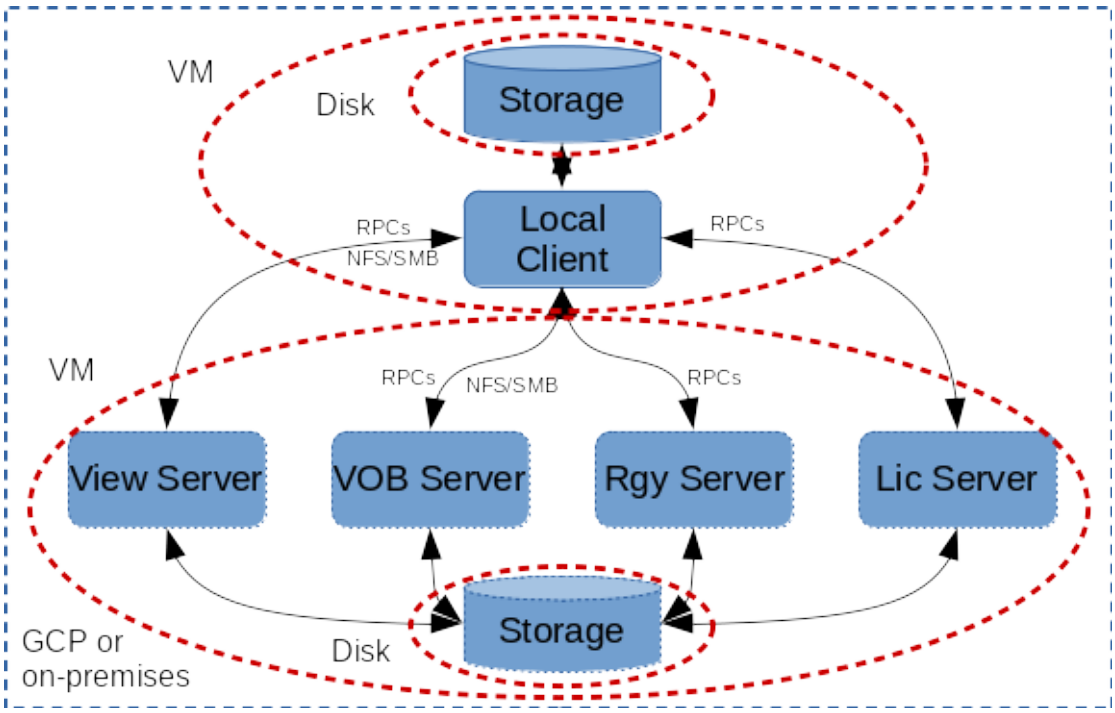

Figure 1: Local client on one machine and servers on another machine

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 GCP 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 GCP 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 GCP in the same zone. 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 GCP and the clients could be in different zones, or even different GCP regions, from the servers. Again, the servers would need to be in the same zone, 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 GCP with a VM instance and its associated disk for the client, another VM instance and its associated disk for the CCRC WAN Server, and another VM instance and its associated disk for the other servers. The CCRC WAN Server VM instance and the other servers VM instance would be in the same zone so that the communication between them would be low latency. The client VM instance could be “anywhere”, e.g., in another GCP region or on the on-premises network, since it is not network latency sensitive.

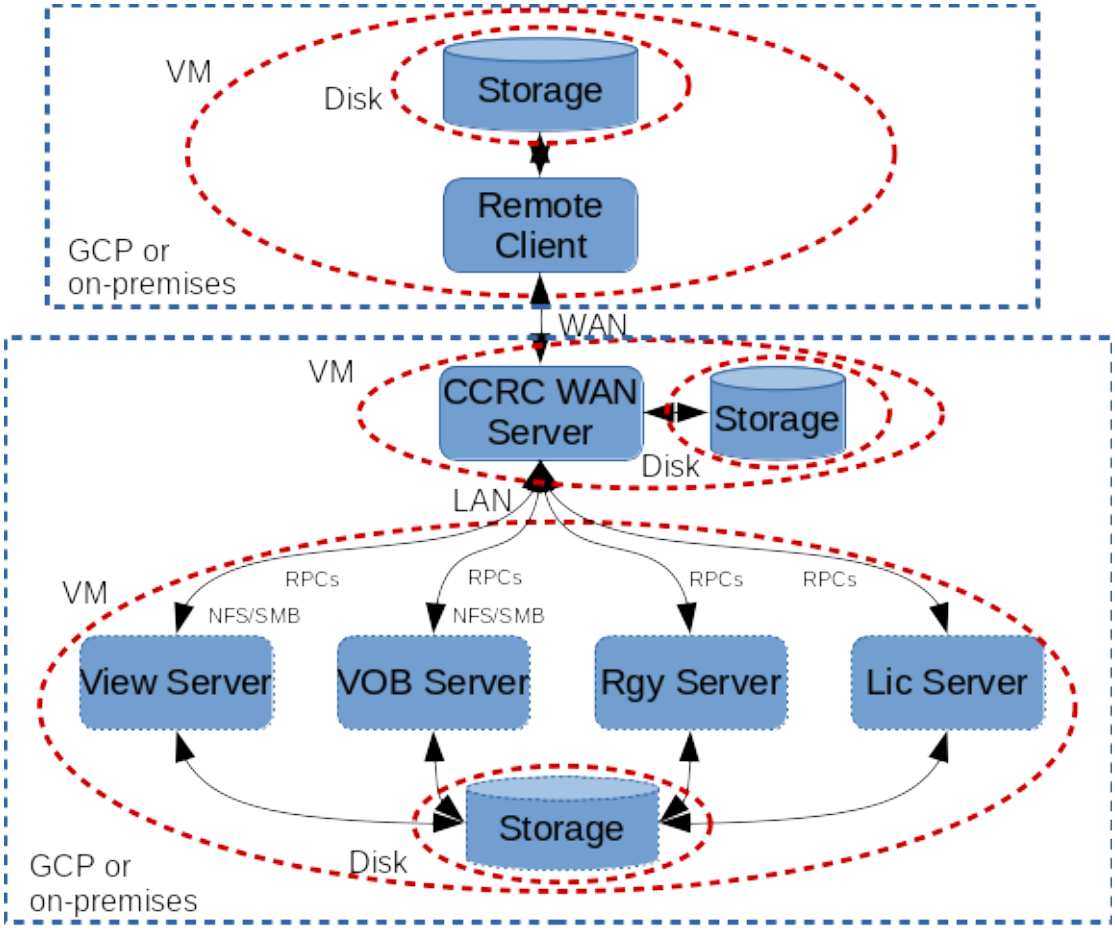

Figure 2: Remote client on one machine and servers on other machines

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

The customer has an existing on-premises HCL VersionVault installation using local view types and wants to add clients and servers in GCP that are synchronized by HCL VersionVault MultiSite. If there are firewall(s) between your on-premises network and GCP (e.g., on your on-premises network and/or on your GCP VPC), 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 GCP where the VOBs are synchronized using HCL VersionVault MultiSite (see Administering HCL VersionVault MultiSite for details).

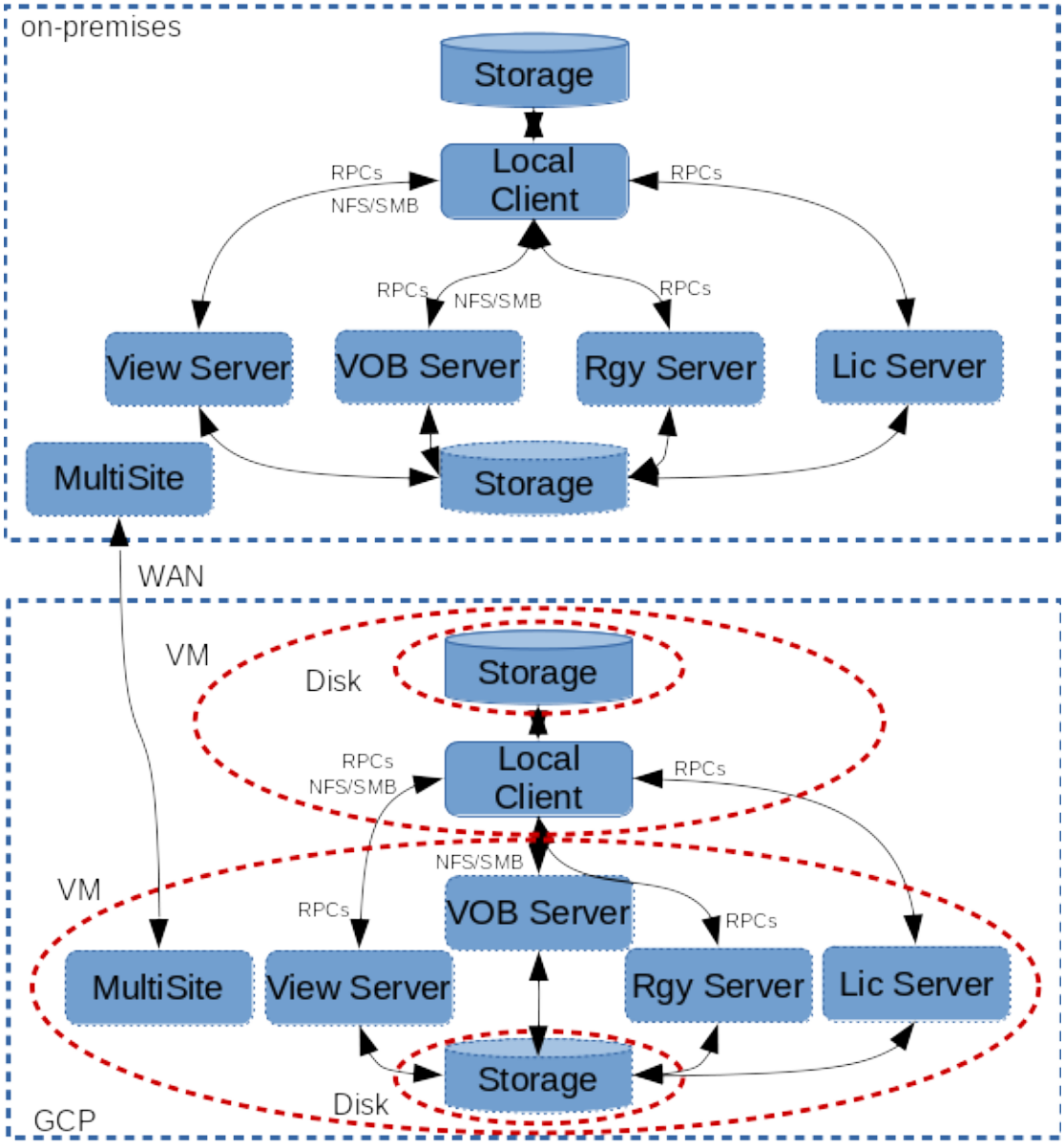

Figure 3: HCL VersionVault MultiSite between on-premises and GCP

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 GCP zone or all in the on-premises network.

For scenarios where network latency is “too high”“, e.g., a hybrid cloud between on-premises and GCP or between VM instances in different GCP 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

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

GCP Documentation - The GCP 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.

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.

What is Cloud Computing - GCP information on cloud computing in general.