

# HCL VersionVault in AWS

**Author:** Rick Kissel

**Date:** 2020-04-30

**Contact:** [howie.bernstein@hcl.com](mailto:howie.bernstein@hcl.com)

**Copyright:** Copyright© HCL Technologies Ltd. 2019, 2020. All Rights Reserved.

## Contents

<b>1</b>	<b>Summary</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Overview of Some AWS Capabilities</b>	<b>2</b>
3.1	Amazon Elastic Compute Cloud	2
3.2	AWS Directory Services	3
3.3	Amazon WorkSpaces	4
<b>4</b>	<b>Plan Your HCL VersionVault Deployment in AWS</b>	<b>4</b>
4.1	HCL VersionVault Servers (VOBs/Views/Registry)	4
4.2	HCL VersionVault Clients	4
<b>5</b>	<b>AWS Considerations</b>	<b>5</b>
5.1	AWS EC2 Instances and AMIs	5
5.2	EC2 instances versus Amazon WorkSpaces	6
5.3	Instance types	6
5.4	VOB and View Storage	6
5.5	Communication between AWS and on-premises networks	7
5.6	Network latency: intra-AZ, intra-region, inter-Region	7
5.7	Security and access control	7
5.8	Active Directory Support	8
5.9	HCL VersionVault MultiSite	8
5.10	Backups	8
5.11	Resilience	8
5.12	Monitoring	8
5.13	Performance	8
5.14	Servers, clients, or both?	9
<b>6</b>	<b>Sample Usage Scenarios</b>	<b>9</b>
6.1	Scenario 1: Local view types (dynamic and snapshot)	9
6.2	Scenario 2: Remote view types (automatic and web)	10
6.3	Scenario 3: Using HCL VersionVault MultiSite between on-premises and AWS	11
6.4	Scenarios common between local and remote view types	12
<b>7</b>	<b>References And More Information</b>	<b>13</b>

# 1 Summary

HCL VersionVault™ can run successfully in the AWS Cloud. AWS provides virtual machines ([EC2 instances](#)) and other services (e.g., [EBS \(Elastic Block Storage\)](#) and [AWS Directory Services](#)) that can be used to run HCL VersionVault servers and clients. AWS EC2 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 AWS.

This paper will discuss various options for configuring your set of AWS 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 AWS EC2 instances. For local clients (dynamic or snapshot views) this will usually require that the clients and the servers they use run in the same Amazon [AZ \(Availability Zone\)](#). For remote clients (automatic or web views) the clients can be separated from the servers they use in different AWS [regions](#) or between AWS and on-premises networks. AWS documentation and tools should be consulted when setting up HCL VersionVault in AWS 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 AWS. Familiarity with HCL VersionVault configuration and administration is assumed.

The paper starts with an overview of some AWS 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 AWS Capabilities

This section summarizes some of the AWS capabilities and terminology that would be useful to HCL VersionVault administrators as they consider using AWS to support their HCL VersionVault installations. Later sections will discuss using these AWS capabilities for specific parts of a HCL VersionVault installation. The [AWS Website](#) provides complete information about AWS and its capabilities. AWS also provides [What is Cloud Computing](#) as an introduction to cloud computing concepts.

### 3.1 Amazon Elastic Compute Cloud

This is an annotated outline of links that follows the information in the [Amazon Elastic Compute Cloud](#) document and provides some annotations and links to other information in the [AWS Documentation](#). Note, there is a [User Guide for Linux Instances](#) and a [User Guide for Windows Instances](#) that are very similar and many of the links below point into the Linux user guide for their information.

- [Amazon Elastic Compute Cloud \(EC2\) Overview](#)

Amazon [EC2](#) provides the computing capacity in the AWS cloud. It provides the following major features.

- [Instances and AMIs](#)
- [Regions and Availability Zones \(AZs\)](#)
- [Root Device Volume](#)
- [AMI \(Amazon Machine Image\)](#)

An AMI provides a template for creating instances.

- [Amazon EC2 Instances](#)

An EC2 instance is a virtual server running a copy of an AMI.

- [Instance Types](#)

The instance type determines the hardware characteristics for your instance, which include compute, memory, and storage capabilities.

- **Storage**

Amazon EC2 provides storage that belongs to EC2 instances (similar to local disk storage) and file storage that is shareable and independent of EC2 instances.

- Amazon **EBS** (Elastic Block Storage)

EBS provides persistent block-level storage volumes for your EC2 instance.

- Amazon EC2 **Instance Store**

Instance store provides temporary block-level storage for your EC2 instance.

- **File Storage**

File storage is similar to network attached storage (NAS) devices.

- Amazon **EFS** (Elastic File System)

EFS provides shareable NFS-based file storage for your EC2 instances.

- Amazon **FSx** (Windows File Server)

FSx provides shareable SMB-based file storage for your EC2 instances.

- **Network and Security**

Amazon EC2 provides many features for networking and security, including the following.

- **Elastic IP Addresses**

Elastic IP addresses are static IPv4 addresses associated with your AWS account that can be used to provide a persistent externally visible addresses for your EC2 instances.

- Amazon **VPC** (Virtual Private Cloud)

AWS provides a [whitepaper](#) that describes the various connection options between a VPC and external networks. In particular, on-premises networks might want to use [AWS Direct Connect](#) to the AWS VPC.

There is a [VPC User Guide](#) that describes [IP addressing](#), [using DNS](#), and [Security](#) that can help with setting up your AWS networking configuration and managing access in the VPC.

- **AWS IAM** (Identity and Access Management)

AWS IAM provides the security controls for policies, roles, etc., to control access to the resources in AWS.

- **Monitoring**

This page provides information on best practices for monitoring in Amazon EC2 and discusses tools such as [Monitoring Your Instances Using CloudWatch](#) and [Automated and Manual Monitoring](#).

## 3.2 AWS Directory Services

This outline follows the [AWS Directory Services](#) document.

- **What is AWS Directory Service?**

This page provides some help in choosing which type of AWS Directory Service you might want to use.

- **AWS Managed Microsoft AD**

This product provides Microsoft AD as a managed service.

- [AWS Active Directory Connector](#)

This product provides a directory gateway so you can redirect directory requests to your on-premises Microsoft AD.

- [AWS Simple Active Directory](#)

This product provides a simple managed directory based on a Samba 4 AD compatible server.

### 3.3 Amazon WorkSpaces

[Amazon WorkSpaces](#) is a product that provides a managed Windows or Linux desktop for your users. You can find more detailed information in the [Amazon WorkSpaces Administration Guide](#).

## 4 Plan Your HCL VersionVault Deployment in AWS

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

Determining key aspects of your HCL VersionVault deployment will provide information that will be useful in making decisions about how AWS 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?
  - AWS?
  - On-premises?
  - AWS 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?

### 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?
- How do you access VOB and view storage from clients?
  - NFS?
  - SMB (CIFS)?
  - Both? To the same storage?
- 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?
- Where will the servers run and where will their storage reside?
  - AWS?
  - On-premises?
  - AWS and on-premises?
- Do you run large clients as “build servers” (multi-user, multi-view clients)?

## 5 AWS Considerations

### 5.1 AWS EC2 Instances and AMIs

AWS EC2 [instances](#) are created from [AMI](#) templates. AWS has public and private AMIs defined that run on 32-bit or 64-bit x86 or on arm64 architectures (on Linux using either para-virtualized or hardware virtual machine [virtualization types](#)). There are AMIs that support Windows and Linux OS versions that HCL VersionVault also supports, e.g., Windows 10 and RHEL 7.5 on x86\_64, so a complete HCL VersionVault installation using those OS versions can be setup in AWS. There are also AMIs that support OS versions that HCL VersionVault does not support, e.g., Amazon Linux, Kali Linux, Gentoo, or OS versions on arm64, so those AMIs cannot be used to create HCL VersionVault client or server instances. You can check the [requirements](#) for HCL VersionVault to see what hardware and OS versions are supported by HCL VersionVault. Once deployed, EC2 instances can be updated to newer OS versions, or even installed with different OS versions that HCL VersionVault supports.

Since customers can create their own [AMI](#) from a running EC2 instance that has been customized, they can create an AMI containing a complete HCL VersionVault client or server image and then use that AMI to create subsequent 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 instance.

An AWS EC2 instance also has [instance metadata and user data](#). You can specify the user data when launching the instance and then that user data can then be accessed by the instance for configuration after it is launched. Using the user data this way lets you build more generic AMIs since the data can be used to configure your instance after it is launched. The link above provides some examples of using the instance metadata and user data in this way.

## 5.2 EC2 instances versus Amazon WorkSpaces

An Amazon [WorkSpace](#) can be used for easy client management, e.g., AWS managed OS updates, simplified instance lifetime management, etc., and it might provide a lower cost than an [EC2](#) instance. It can be “always on” or it can “auto stop” after a specified period of inactivity. However, an Amazon WorkSpace has its own managed kernel and so a HCL VersionVault installation that includes the MVFS kernel module will not persist when the WorkSpace restarts.

An EC2 instance is started from an AMI, which can be supplied by the customer. HCL VersionVault, including the MVFS kernel module, can be installed in the AMI so each EC2 instance that uses that AMI would already have HCL VersionVault installed and running. Note, that installation of HCL VersionVault would persist across restarts of the EC2 instance.

An Amazon WorkSpace may be an option for HCL VersionVault remote web view or snapshot view clients or HCL VersionVault servers, since they don’t need the MVFS kernel module. HCL VersionVault dynamic view clients and automatic view clients require the MVFS kernel component for their operation so it might be more appropriate to use an EC2 instance.

For either Amazon WorkSpaces or EC2 instances the customer could supply a post-startup script that would install HCL VersionVault (e.g., using an IM silent install command line with a response file), including the MVFS kernel module (note, on a Windows instance this will require a reboot), and perhaps do other configuration as well. This isn’t necessary on an EC2 instance that used an AMI that included the HCL VersionVault installation. Such a script would be necessary for an Amazon WorkSpace to use dynamic or automatic view clients and it may slow down the Amazon WorkSpace startup time which might negate some of the benefits of using an Amazon WorkSpace.

## 5.3 Instance types

AWS provides many [instance types](#) that can be used for [EC2 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 AWS [general purpose](#) instance types are normally suitable for HCL VersionVault. Some heavily used VOB and view servers may benefit from the [memory optimized](#) instance types because the HCL VersionVault database used by these servers can benefit from more (or faster) memory. Many clients and other servers may have a more “bursty” load so [burstable performance](#) instance types might be a better choice. There are also some bare metal instance types (although they are very large) that could be appropriate for very heavily used VOB or view servers.

AWS provides various [CloudWatch Metrics](#) that you can use to evaluate your instance types to help you decide on the most cost effective instance types to use for your environment. Note, you can [change the instance type](#) for an EC2 instance by stopping it, changing the instance type in the EC2 management console, and restarting it. Thus, as your needs evolve (or you’ve made a mistake), you can change the instance type to a more appropriate one without losing your EC2 instance.

## 5.4 VOB and View Storage

Since [EFS](#) is oriented towards file sharing among Linux EC2 instance (using NFSv4) and [FSx](#) is oriented towards file sharing among Windows EC2 instances (using SMB), neither of them may be suitable for a mixed Linux and Windows environment. However, they could be used somewhat like NAS storage in a homogeneous environment for VOB and view storage. [EBS](#) storage is the “local disk” persistent storage for EC2 instances, either Windows or Linux, so is more appropriate for HCL VersionVault VOB and view storage. This does mean that each HCL VersionVault server needs to export its storage (e.g., through SMB and/or NFS) so that remote clients can access it.

## 5.5 Communication between AWS and on-premises networks

AWS provides a [whitepaper](#) that describes the various connection options between a VPC and external networks. In particular, on-premises networks might want to use [AWS Direct Connect](#) to the AWS VPC. You might also want to consider using [Elastic IP Addresses](#) to provide permanent, externally accessible IP addresses for some of your EC2 instances, especially HCL VersionVault servers. These Elastic IP Addresses can also be used for internal communication since AWS is “smart enough” to route internally even though the IP addresses are visible externally. There is a [VPC User Guide](#) that describes [IP addressing](#) as well as [using DNS](#) and can help with setting up your AWS networking configuration. Note, the clients and servers for HCL VersionVault need to be protected from outside (public) access by a firewall or other similar technology. The VPC User Guide contains a section on [Security](#) that describes the AWS capabilities for managing access in a VPC.

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 AWS instances will not provide acceptable performance for dynamic views.

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

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

- [intra-AZ](#) network latency should be suitable for any HCL VersionVault communication
- [intra-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.
- [inter-region](#) or AWS to customer on-premises 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 and access control

General OS-level access control across EC2 instances is described in [Controlling EC2 OS-Level Access](#), which covers both Linux and Windows instances and recommends ways to manage users and their access. For HCL VersionVault, probably the [AWS Managed Microsoft AD](#) or the [AWS Active Directory Connector](#) would be most appropriate since they can be used for defining users and groups on both Windows and Linux. Of course, the customer can set up their own infrastructure (e.g., Microsoft AD or NIS) on their own EC2 instances and manage OS users and groups themselves, just as they would in an on-premises network.

On Linux, individual users can be created on each instance if necessary, see [Managing User Accounts on Your Linux Instance](#). This might be suitable for a small HCL VersionVault setup and would avoid using a centralized directory service.

For HCL VersionVault within AWS you need to make sure that the AWS [Windows](#) or [Linux](#) Security Groups used by your EC2 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 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 instances. 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 between the clients and the servers.



## 5.8 Active Directory Support

AWS provides a number of [AWS Directory Services](#), including [AWS Managed Microsoft AD](#), among others. AWS documentation provides [some help for choosing](#) the type of directory service you might want to use. HCL VersionVault has no special requirements when using AD in AWS. The same considerations that are needed for AD on-premises also apply to AD in AWS. 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.

## 5.9 HCL VersionVault MultiSite

[HCL VersionVault MultiSite](#) can be used in AWS, and might be needed if dynamic view access is needed and clients and the VOB servers and storage for those VOBs that the clients access are in different AWS regions or are split between AWS and the on-premises network. This is because the AWS inter-region network latency or the network latency between AWS and an on-premises network is high enough that dynamic views would perform poorly when accessing the "remote" servers. HCL VersionVault MultiSite is designed to alleviate this problem.

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 MultiSite 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 AWS HCL VersionVault installations.

## 5.10 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 EBS [Linux Snapshots](#) or [Windows Snapshots](#) of the storage on particular servers could be used as part of your backup implementation. For VOB data, HCL VersionVault MultiSite to servers in a different AWS [Region](#) could be part of your backup solution since this avoids a single point of failure in the AWS infrastructure because AWS regions are relatively independent.

## 5.11 Resilience

Amazon provides lots of information on resilience considerations including [AWS Direct Connect Resiliency Recommendations](#) and a new whitepaper [Achieving Operational Resilience in the Financial Sector and Beyond](#).

## 5.12 Monitoring

[AWS Monitoring](#) provides a number of services including details described in [Linux Monitoring](#) and [Windows Monitoring](#). These can be used to gauge the health of the various servers and clients running in AWS. They can also provide notifications if various configured thresholds are exceeded.

## 5.13 Performance

A few performance considerations have been mentioned in previous sections, so those should be kept in mind when configuring your AWS environment. There are many options in AWS for configuring the the number of cpus, memory size, network bandwidth, dedicated vs. shared instances, etc. (see [instance types](#)).

In general, HCL VersionVault performance in AWS 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 \(pt 2\)](#).

## 5.14 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 AWS [AZ](#) 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 between AZs in the same region may or may not be acceptable for any HCL VersionVault communications; it should be measured to determine its acceptability. The network latency between AWS regions and between AWS 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 AWS 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 AWS for HCL VersionVault. These scenarios assume the customer has suitable access to clients and servers running in AWS 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 AWS. Due to the expected network latency between the customer’s on-premises network and AWS, 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 AWS and the on-premises network. The isolated subset of clients and servers should be moved to a single [AZ](#) in AWS 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 AWS with an EC2 instance and its associated EBS for the client and another EC2 instance and its associated EBS for the servers. These two EC2 instances 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.

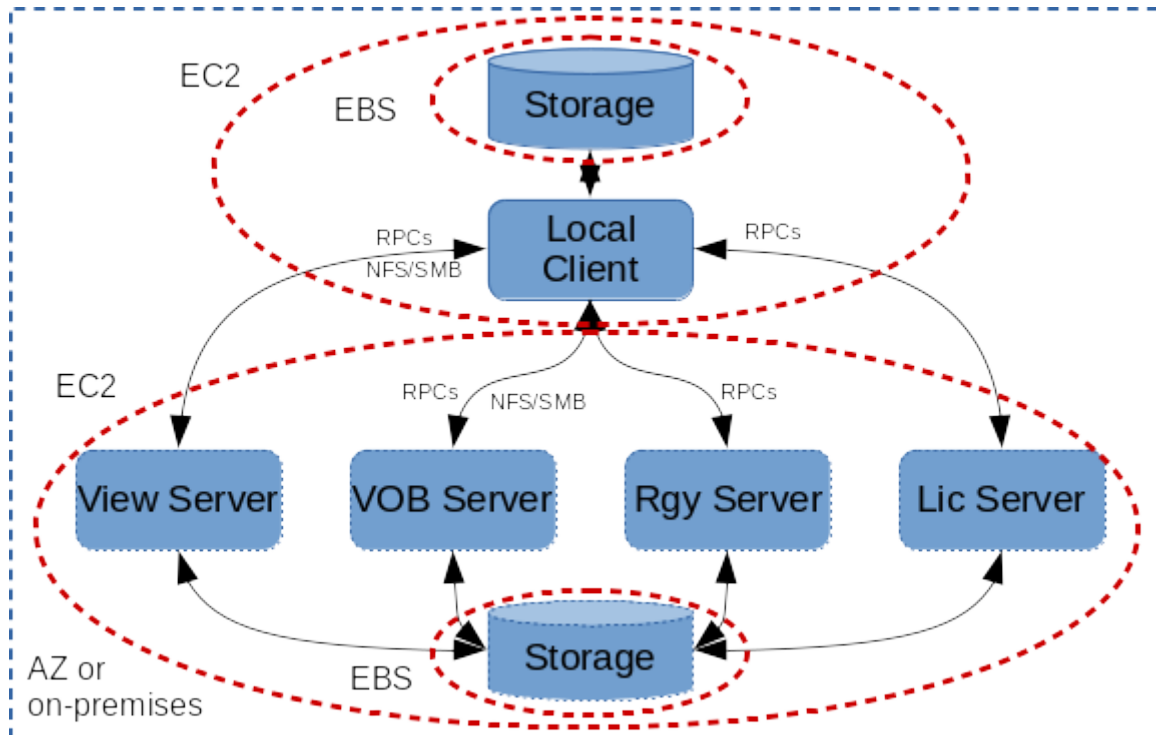


Figure 1: Local client on one machine and servers on another machine

## 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 AWS 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 AWS 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 AWS 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 AWS and the clients could be in different AZs, or even different AWS 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 AWS with an EC2 instance and its associated EBS for the client, another EC2 instance and its associated EBS for the CCRC WAN Server, and another EC2 instance and its associated EBS for the other servers. The CCRC WAN Server EC2 instance and the other servers EC2 instance would be in the same AZ so that the communication between them would be low latency. The client EC2 instance could be “anywhere”, e.g., in another AWS region or on the on-premises network, since it is not network latency sensitive.

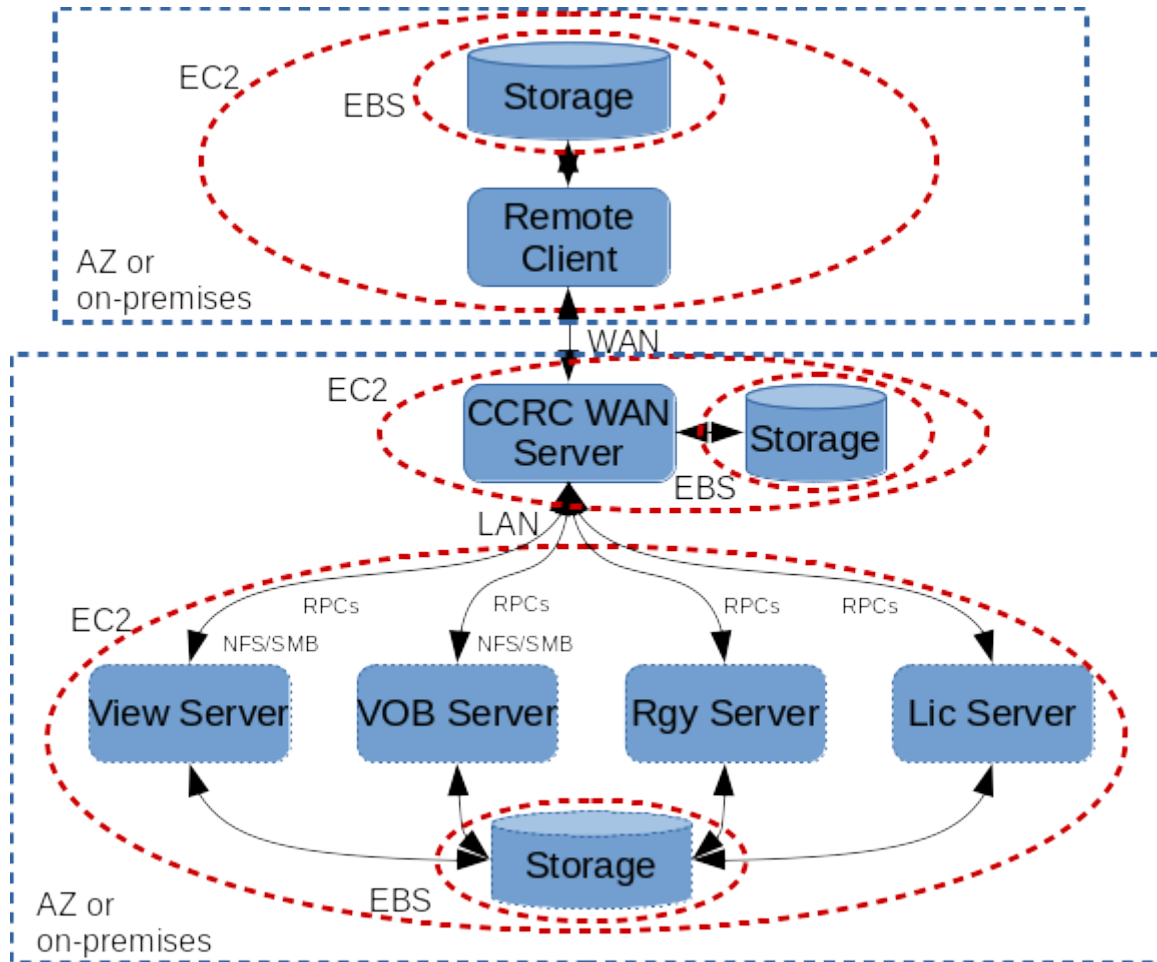


Figure 2: Remote client on one machine and servers on other machines

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

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

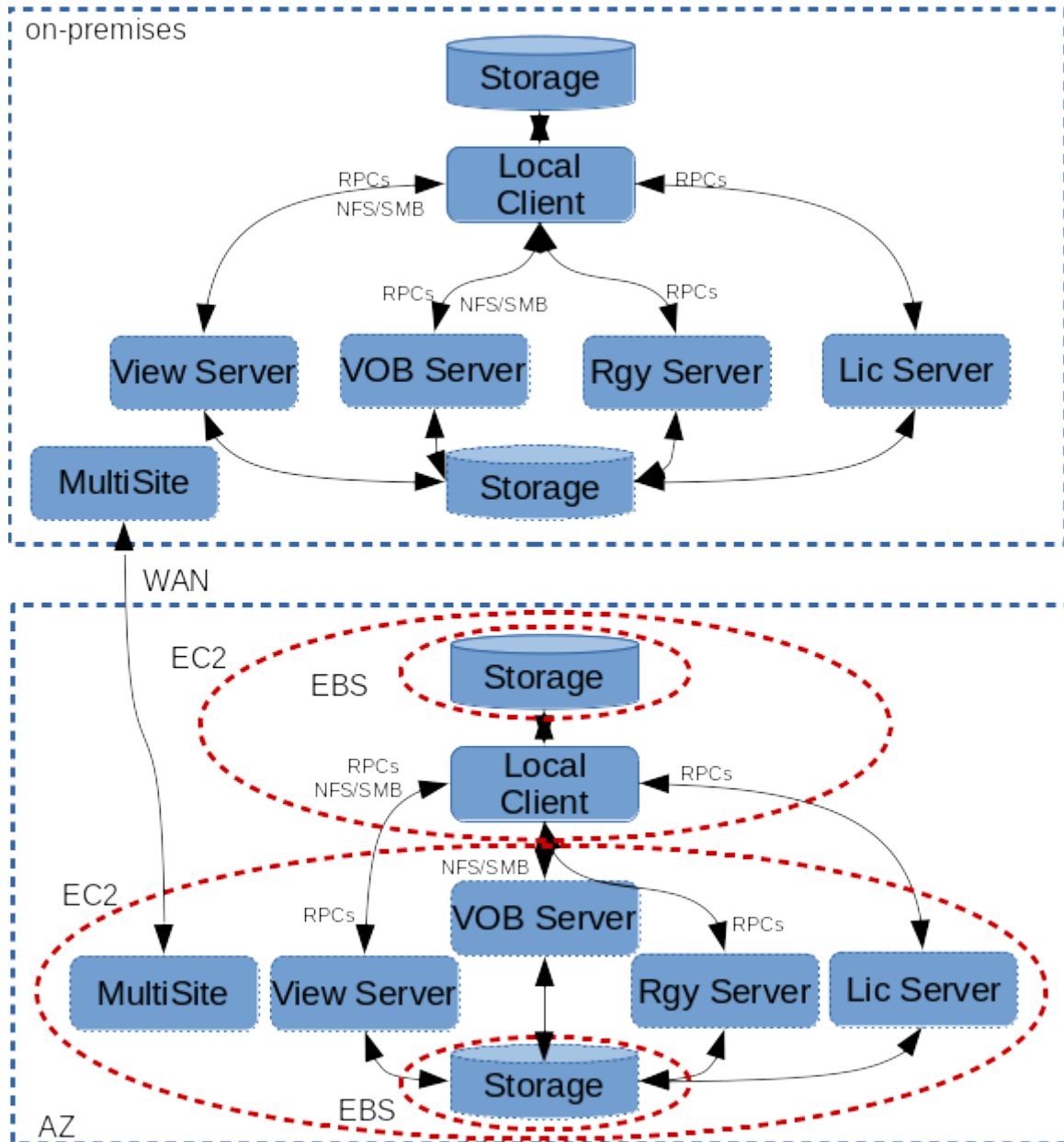


Figure 3: HCL VersionVault MultiSite between on-premises and AWS

## 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 AWS 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 AWS or between EC2 instances in different AWS 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

[AWS Website](#) - The main AWS website from which everything AWS related can be found.

[AWS Documentation](#) - The AWS 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](#).

[What is Cloud Computing](#) - AWS information on cloud computing in general.