App Volumes Configuration

Omnissa Horizon

Table of contents

App Volumes Configuration 4

Installation and initial configuration 5

- Rebranding changes in Omnissa Horizon products 5

Using SQL Always on with App Volumes 6

- Create a Windows server failover cluster in each site 6
- Install SQL server stand-alone in all VMs 7
- Create the App Volumes databases and enable availability groups for the clusters 9
- Create Always On availability groups for App Volumes databases 10
- Configure cluster quorum settings 12
- Install App Volumes to use a highly available database 14

Multi-instance configuration 16

- Install and configure App Volumes managers 16
- Setup package replication 17
- Configure a not attachable NFS datastore for site 1 and site 2 instances 17
- Mark shared datastore as read-only on target instances 19
- Create storage groups 19

Additional configuration options for Writable Volumes 21

- Allowing end users to see the size of Writable Volumes 21
- End-user viewing of Writable Volume space 21
- Expanding Writable Volumes 22

Understand packages and packaging for day 2 operations 23

- Package templates 23
- Packages at scale 23
- Recommended practices for packages in production environments 23
- Packaging applications recommended practices 24
- Updating and assigning updated packages recommended practices 24
- Performance testing for packages 27
- ThinApp integration with packages 27
- Microsoft Office application packages 27
- Office plug-ins and add-ons 27
- RDSH integration with packages 27
- Application suitability for packages 28
- Applications that work best in the golden image 28
- Applications whose components are not well understood 28
- App Volumes Agent altitude and interaction with other mini-filter drivers 28

Other special considerations 29

Recommended practices for production environments 30

Golden image optimization 30

Desktop pool recommendations 30

Summary and additional resources 31

Additional resources 31

Changelog 31

Author and contributors 31

Feedback 31

App Volumes Configuration

This chapter is one of a series that make up the Omnissa Workspace ONE and Horizon Reference Architecture, a framework that provides guidance on the architecture, design considerations, and deployment of Omnissa Workspace ONE and Omnissa Horizon solutions. This chapter provides information about common configuration and deployment tasks for App Volumes.

This includes detailed instructions for configuring App Volumes across multiple sites, implementing redundancy both within and across sites. As is described in the Multi-site design section of App Volumes Architecture, the recommended deployment option for App Volumes across multiple sites is to use separate databases.

The procedures in this guide create a setup in which the App Volumes database can fail over automatically within each site. Site 1 and Site 2 have separate databases, but during a failover, users at Site 1 will be able to use replicated App Volumes packages in Site 2, as long as the user entitlements are also replicated using multi-instance management. See Multi-instance management for more information.

Installation and initial configuration

Refer to the App Volumes 4 Install Guide for full installation procedures. The prerequisites are covered in more detail in the System Requirements section. The App Volumes Administration guide details the initial configuration.

After installation is complete, you must perform the following tasks to start using App Volumes:

- Complete the App Volumes initial configuration wizard (https://avmanager).
- Install the App Volumes Agent on one or more clients and point the agent to the App Volumes Manager address (load-balanced address).
- Select a clean packaging system and create an application package. See Working with Applications in App Volumes and Working with Packages for instructions.
- Assign the package to a test user and verify it is connecting properly.
- Assign a Writable Volume to a test user and verify it is connecting properly.

For better performance, it is recommended to use a local vCenter account instead of an active directory account when configuring a vCenter Server as a machine manager.

- For details on how to create a local user account, see Managing Local User Accounts in vCenter Server.
- Information on the permissions required for a vCenter account are detailed in the App Volumes Manager console, at the bottom of the Configuration, machine managers section.

See Configure and Register the Machine Manager for more information on registering a vCenter Server as a machine manager.

Refer to product documentation Deploy App Volumes for Amazon AppStream 2.0 for automated deployment of App Volumes from AWS Marketplace and Deploy App Volumes Manager through Azure Marketplace for automated deployment of App Volumes from Azure Marketplace.

Rebranding changes in Omnissa Horizon products

As part of Omnissa’s launch as an independent company, significant updates to the Horizon 8, App Volumes and Dynamic Environment Manager product family, starting with the 2412 release, are documented in the Kb article - Rebranding Changes in Omnissa Horizon Products (6000681).

Using SQL Always on with App Volumes

This design uses separate App Volumes database instances for each site. With this option, you can use SQL Always on availability groups within each site to achieve local high availability of the database.

To use this setup, perform the procedures in the following order:

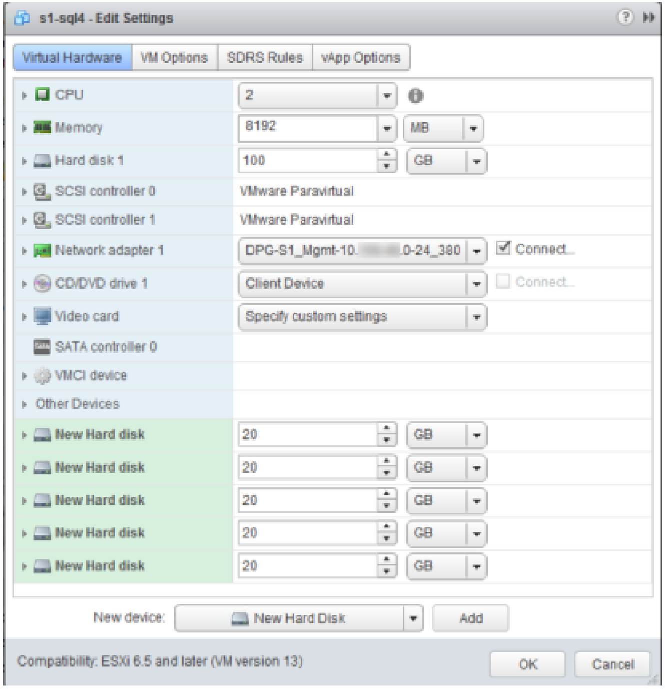
1. Create a Windows Server Failover Cluster in each site
2. Install SQL Server stand-alone in all VMs
3. Create the App Volumes databases and Enable Availability groups for the glusters
4. Create Always on Availability groups for App Volumes databases
5. Configure Cluster Quorum settings
6. Install App Volumes to use a highly available database

Create a Windows server failover cluster in each site

A failover cluster is a group of VMs that have the same software installed on them and work together as one instance to provide high availability for a service, such as a Microsoft SQL Server database. If a VM, or cluster node fails, another node in the cluster begins to provide the service.

To create a Windows Server Failover Clustering (WSFC) cluster:

1. On two ESXi hosts in Site 1, create and configure the two VMs that will be used as a clustered instance of Microsoft SQL Server, and then do the same in Site 2.
    a. For this reference architecture, we used the Windows Server 2019 Standard operating system in the VMs that run SQL Server.
    b. In this example, we named the VMs as follows: At Site 1, we created VMs named s1-sql4 and s1-sql5 on separate ESXi hosts. At Site 2, we created VMs named s2-sql4 and s2-sql5.
    c. Set up each VM in the same way, with a total of five 20-GB hard disks: one disk for the Windows OS and five disks for the various types of SQL Server data.



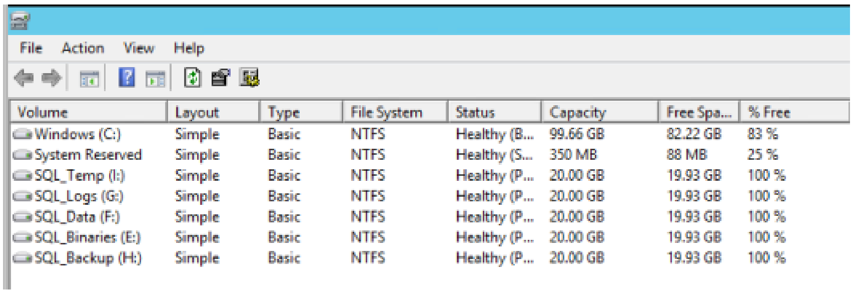
s1-sql4 - Edit Settings

CPU	2	
Memory	8192	MB
Hard disk 1	100	GB
SCSI controller 0	VMware Paravirtual	
SCSI controller 1	VMware Paravirtual	
Network adapter 1	DPG-S1_Mgmt-10.[redacted]0-24_380	☑ Connect...
CD/DVD drive 1	Client Device	□ Connect...
Video card	Specify custom settings	
SATA controller 0		
VMCI device		
Other Devices		
New Hard disk	20	GB
New Hard disk	20	GB
New Hard disk	20	GB
New Hard disk	20	GB
New Hard disk	20	GB

New device: New Hard Disk

Compatibility: ESXi 6.5 and later (VM version 13)

Inside the OS, the disks are mapped to drive letters as follows.



Volume	Layout	Type	File System	Status	Capacity	Free Spa...	% Free
Windows (C:)	Simple	Basic	NTFS	Healthy (B...	99.66 GB	82.22 GB	83 %
System Reserved	Simple	Basic	NTFS	Healthy (S...	350 MB	88 MB	25 %
SQL_Temp (I:)	Simple	Basic	NTFS	Healthy (P...	20.00 GB	19.93 GB	100 %
SQL_Logs (G:)	Simple	Basic	NTFS	Healthy (P...	20.00 GB	19.93 GB	100 %
SQL_Data (F:)	Simple	Basic	NTFS	Healthy (P...	20.00 GB	19.93 GB	100 %
SQL_Binaries (E:)	Simple	Basic	NTFS	Healthy (P...	20.00 GB	19.93 GB	100 %
SQL_Backup (H:)	Simple	Basic	NTFS	Healthy (P...	20.00 GB	19.93 GB	100 %

2. In preparation for creating the failover cluster, choose a cluster name and obtain a corresponding static IP address for the cluster in Site 1 and in Site 2.

For example, in this reference architecture, we use s1-sqlclust-2 and s2-sqlclust-2 for the cluster names.

3. To set up failover clustering on the SQL Server VMs, user Server Manager on each of the four VMs (two in Site 1 and two in Site 2), and use the Windows Server Add Roles and Features Wizard to add the Failover Clustering feature.

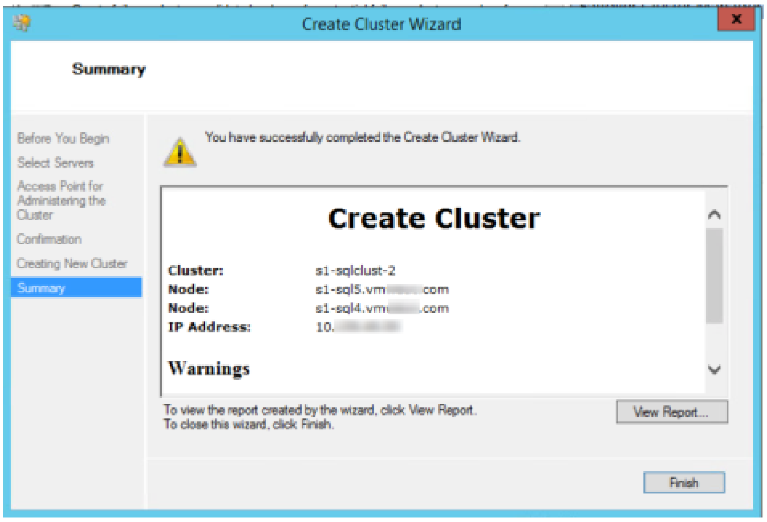
In the wizard, select to add the following features:

- .NET Framework 3.5 features
- Failover Clustering, including adding the Failover Clustering tools

For detailed instructions, see the Microsoft blog post Installing the Failover Cluster Feature and Tools in Windows Server 2012. One of the tools installed is the Failover Cluster Manager, which you will use for many of the steps that follow.

4. Use the Failover Cluster Manager to create a WSFC cluster that includes two SQL Server VMs in Site 1, and then do the same in Site 2.

For detailed instructions, see Create a Failover Cluster. The failover cluster for Site 1 includes the VMs s1-sql4 and s1-sql5. The cluster for Site 2 includes the VMs s2-sql4 and s2-sql5.



You have successfully completed the Create Cluster Wizard.

Create Cluster

Cluster: s1-sqlclust-2
Node: s1-sql5.vm[redacted].com
Node: s1-sql4.vm[redacted].com
IP Address: 10.[redacted]

Warnings

To view the report created by the wizard, click View Report.
To close this wizard, click Finish.

Install SQL server stand-alone in all VMs

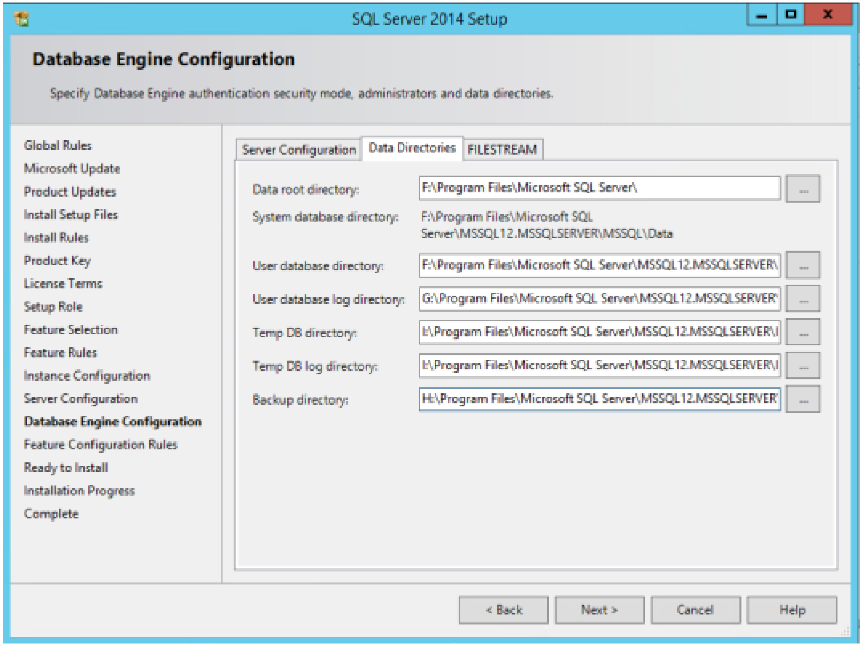
You install SQL Server Stand-Alone on each VM, rather than creating a SQL Server failover cluster. You can stop after the 1st step (i.e.) if you choose to install only a single SQL server instance in your environment and create the App Volumes database. For a high availability of the App Volumes database, you use stand-alone installations and then create Always On availability groups to achieve failover within a site.

To install SQL Server:

1. In Site 1, on the first SQL Server VM, complete the SQL Server installation wizard, using the following guidelines:
   a. Installation page – Select New SQL Server stand-alone installation.
   b. Feature Selection page – Select Database Engine Services and Management Tools - Basic.
   c. Instance Configuration page – Select Default instance. The instance ID is MSSQLSERVER.
   d. Server Configuration page – The startup type is Automatic for the SQL Server Agent and SQL Server Database Engine.
   e. Database Engine Configuration page – On the Server Configuration tab, select Mixed Mode (SQL Server authentication and Windows authentication), and enter credentials for a user account that will be part of the SQL

Server administrators.

On the Data Directories tab, for each item, select the disk that you created for that type of file during VM creation. Use the following screenshot as a guide:



SQL Server 2014 Setup

Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Data root directory: F:\Program Files\Microsoft SQL Server\

System database directory: F:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Data

User database directory: F:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\

User database log directory: G:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER'

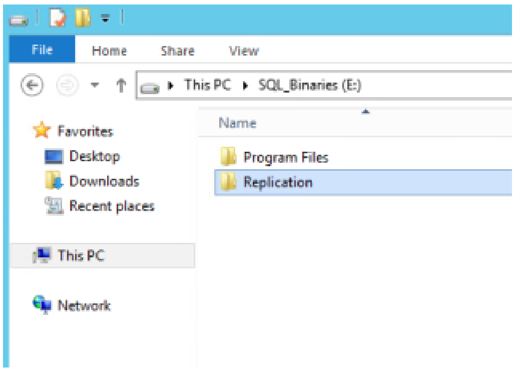
Temp DB directory: I:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\I

Temp DB log directory: I:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\I

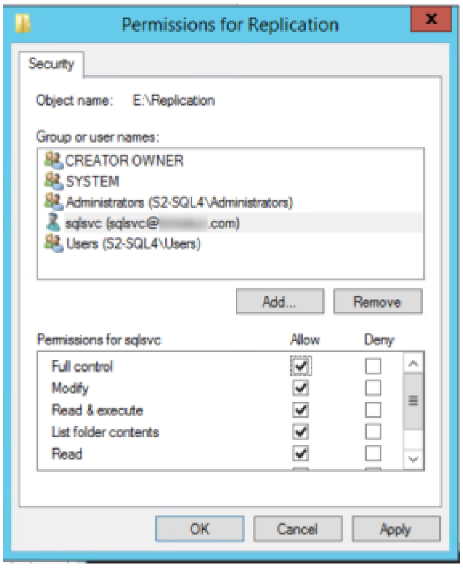
Backup directory: H:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER'

For more information about the setup wizard, see Install SQL Server from the Installation Wizard (Setup).

2. Repeat Step 1 to install SQL Server on the other VM in Site 1 and on the VMs in Site 2.
3. Create a shared folder on the first SQL Server VM in each site; for this example, the VMs are s1-sql4 and s2-sql4).
   a. Create a folder named Replication on the SQL_Binaries (E:) drive. You created this drive during Step 1 of Create a Windows Server Failover Cluster in Each Site.



   b. Give the SQL service account full permissions on the folder.



Permissions for Replication

Object name: E:\Replication

Group or user names:

- CREATOR OWNER
- SYSTEM
- Administrators (S2-SQL4\Administrators)
- sqlsvc (sqlsvc@[redacted].com)
- Users (S2-SQL4\Users)

Permissions for sqlsvc	Allow	Deny
Full control	☑	□
Modify	☑	□
Read & execute	☑	□
List folder contents	☑	□
Read	☑	□

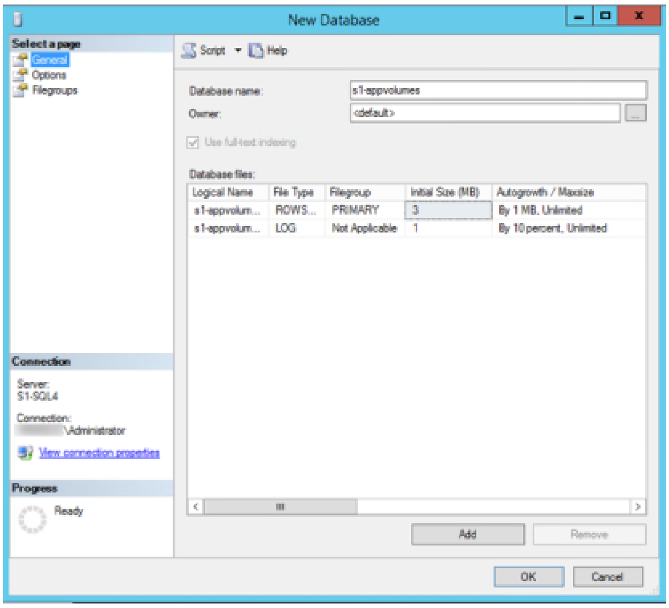
Create the App Volumes databases and enable availability groups for the clusters

This section provides detailed instructions for creating a highly available database, but does not give recommendations regarding the sizing of the database for various sizes of deployments. For information about database sizing, see App Volumes Database Best Practices.

To create the databases:

1. On first SQL Server VM in each site, use Microsoft SQL Server Management Studio to create an App Volumes database.
   a. For this example, the VMs are s1-sql4 and s2-sql4.
   b. Log in to the Management Studio as the sysadmin or as a user account with sysadmin privileges.
   c. Connect to the SQL Server instance on the VM, right-click the Databases folder, and select New Database.
   d. Enter a database name and click OK.

For example, for Site 1, we named the database s1-appvolumes. For Site 2, we named the database s2-appvolumes.



New Database

Database name: s1-appvolumes

Owner: <default>

☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
s1-appvolum...	ROWS...	PRIMARY	3	By 1 MB, Unlimited
s1-appvolum...	LOG	Not Applicable	1	By 10 percent, Unlimited

Connection

Server:
S1-SQL4

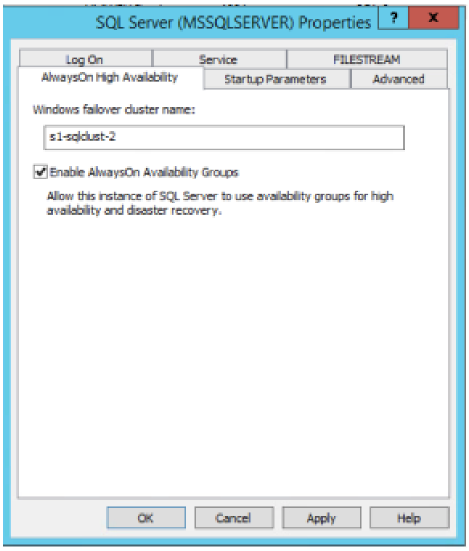
Connection:
[redacted]\Administrator

Progress

Ready

2. Open SQL Server Configuration Manager and enable Always On availability groups for the Windows Failover cluster in Site 1 and Site 2.

For instructions, see Enable or disable Always On availability group feature (SQL Server).


SQL Server (MSSQLSERVER) Properties

Windows failover cluster name:

s1-sqlclust-2

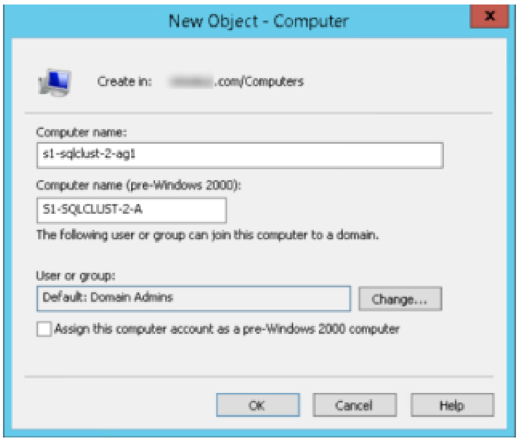
☑ Enable AlwaysOn Availability Groups

Allow this instance of SQL Server to use availability groups for high availability and disaster recovery.

The cluster names are the names you created in Step 2 of Create a Windows Server Failover Cluster in Each Site.

3. In each site, on the domain controller, open Active Directory Users and Groups, and create a new Computer object for the availability group.

For this example, for Site 1, the AD computer name for the group is s1-sqlclust2-ag1. For Site 2, the name is s2-sqlclust2-ag1.


New Object - Computer

Create in: [redacted].com/Computers

Computer name:

s1-sqlclust-2-ag1

Computer name (pre-Windows 2000):

S1-SQLCLUST-2-A

The following user or group can join this computer to a domain.

User or group:

Default: Domain Admins

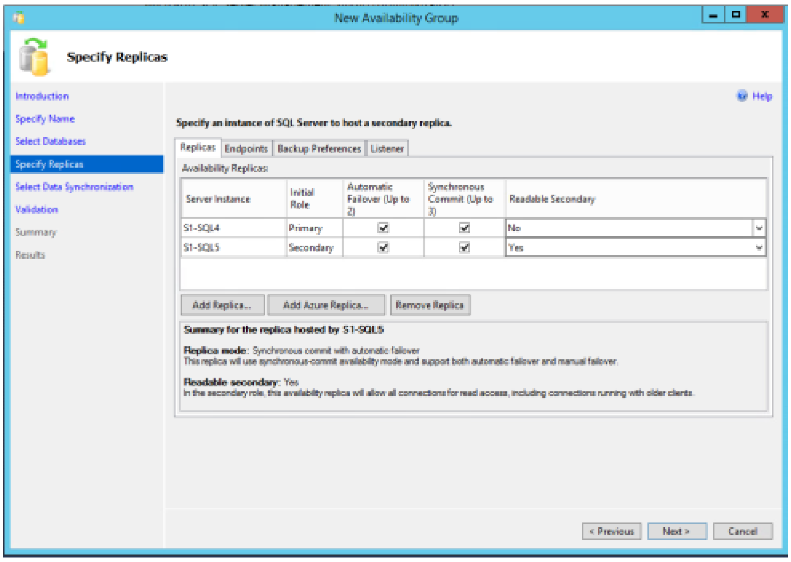
□ Assign this computer account as a pre-Windows 2000 computer

Create Always On availability groups for App Volumes databases

In this procedure, you create an Always On availability group, adding the SQL Server stand-alone instances from Site 1 as the primary replica and secondary replicas. You then do the same for Site 2, so that each site has its own Always On availability group to achieve automatic failover within each site (but not across sites).

To create the availability groups:

1. On the first SQL Server VM in Site 1, open the Management Studio, right-click Always On High Availability in the left pane, and select New Availability Group Wizard.
2. Complete the New Availability Group wizard, using the following guidelines:
   a. Specify Name page – Use the name of the AD computer object you just created. For this example, the name is s1-sqlclust2-ag1 for Site 1.
   b. Select Databases page – Select the check box for the local database, which is s1-appvolumes in this example.
   c. Specify Replicas > Replicas tab – The first SQL Server VM should already be listed. Click Add Replica to connect to and add the second SQL Server VM from this site.



Specify Replicas

Specify an instance of SQL Server to host a secondary replica.

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 2)	Synchronous Commit (Up to 3)	Readable Secondary
S1-SQL4	Primary	☑	☑	No
S1-SQL5	Secondary	☑	☑	Yes

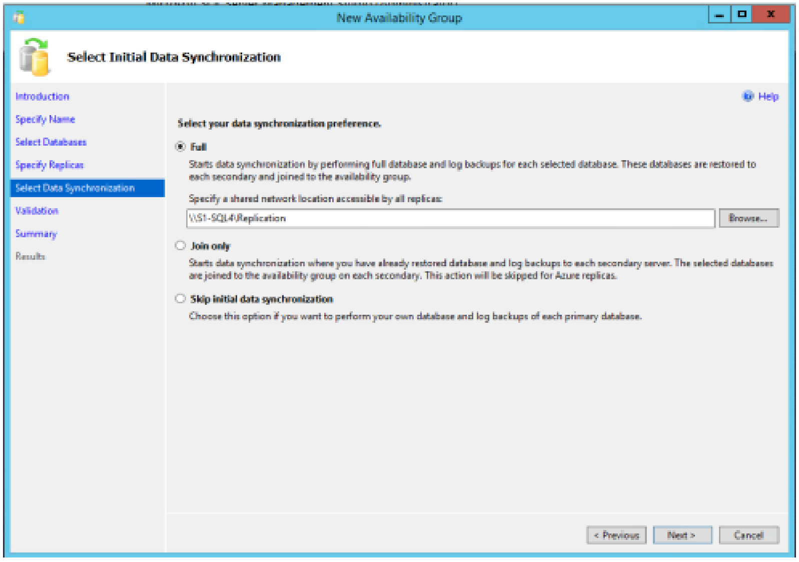
Summary for the replica hosted by S1-SQL5

Replica mode: Synchronous commit with automatic failover
This replica will use synchronous-commit availability mode and support both automatic failover and manual failover.

Readable secondary: Yes
In the secondary role, this availability replica will allow all connections for read access, including connections running with older clients.

Important: Select the Synchronous Commit and Automatic Failover check boxes. For the Readable Secondary setting, select No for the primary instance and Yes for the secondary instance.

d. Select Data Synchronization page – Select Full, and specify the shared Replication folder that you created in Step3, Install SQL Server Stand-Alone in All VMs.



Select Initial Data Synchronization

Select your data synchronization preference.

● **Full**
Starts data synchronization by performing full database and log backups for each selected database. These databases are restored to each secondary and joined to the availability group.

Specify a shared network location accessible by all replicas:
\\S1-SQL4\Replication

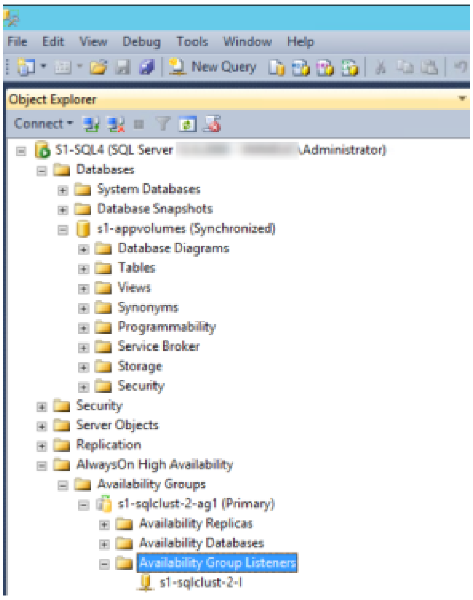
○ **Join only**
Starts data synchronization where you have already restored database and log backups to each secondary server. The selected databases are joined to the availability group on each secondary. This action will be skipped for Azure replicas.

○ **Skip initial data synchronization**
Choose this option if you want to perform your own database and log backups of each primary database.

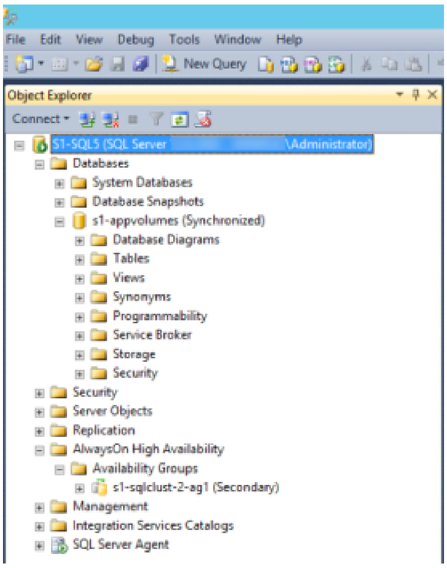
This share is used to synchronize the database on the secondary replica with that on the primary.

After you complete these pages, the validation page, the summary page, and the results page take you through the process of creating the availability groups and listeners, and adding the replicas. On the results page, you can see that the s1-appvolumes database is synchronized between both SQL servers.

When the process is complete, you can view the new availability groups using the Management Studio. The following screenshot shows the availability group with the primary replica on the s1-sql4 VM.



The following screenshot shows the availability group with the secondary replica on the s1-sql5 VM.



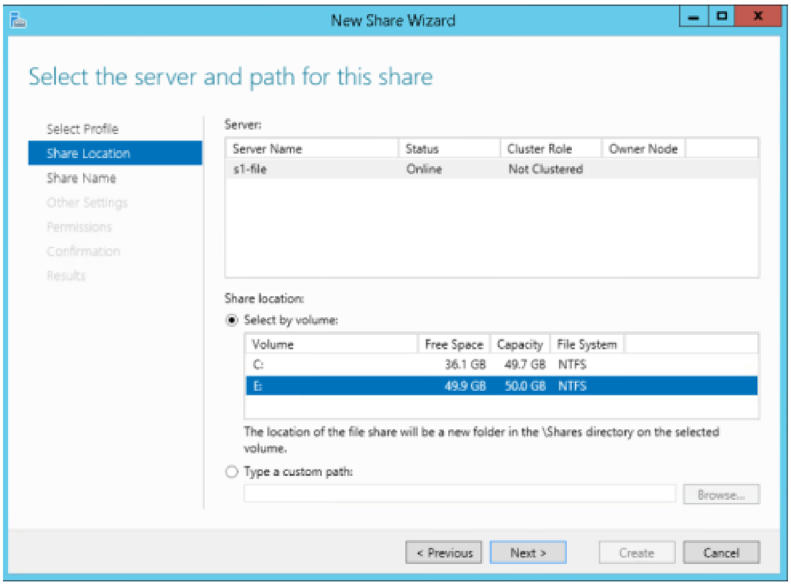
3. Repeat Steps 1 and 2 for Site 2.

Configure cluster quorum settings

At this point, you will configure cluster quorum settings to use a file share witness. Each element in a cluster can cast one “vote” to determine whether the cluster can run. Because you have two nodes in a cluster and you need an odd number of voting elements, create a file share quorum witness, which will cast the third vote. A file share witness is recommended when you need to consider multi-site disaster recovery with replicated storage.

To configure cluster quorum settings:

1. On a file server in each site, use Server Manager to open and complete the New Share wizard and create an SMB share, using the following guidelines:
    a. Select Profile page – Select SMB Share – Quick.
    b. Share Location page – For the share location, choose Select by volume, and select the drive.



New Share Wizard

Select the server and path for this share

- Select Profile
- Share Location
- Share Name
- Other Settings
- Permissions
- Confirmation
- Results

Server:

Server Name	Status	Cluster Role	Owner Node
s1-file	Online	Not Clustered	

Share location:

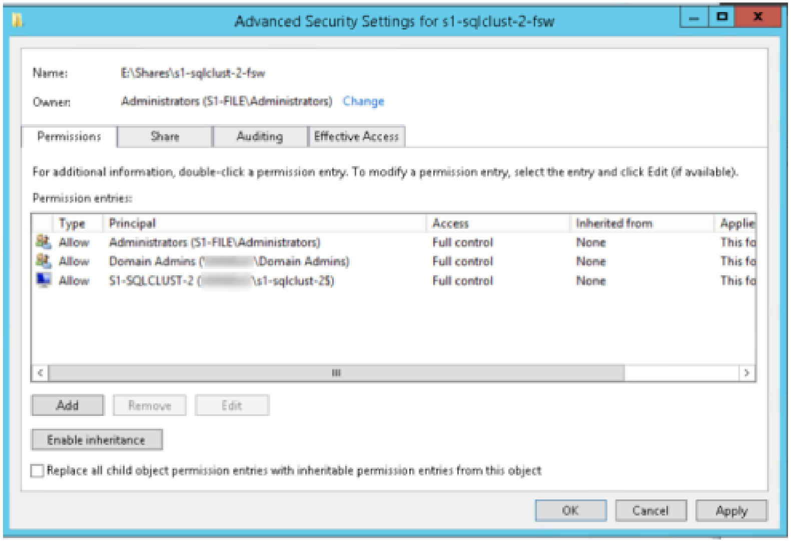
● Select by volume:

Volume	Free Space	Capacity	File System
C:	36.1 GB	49.7 GB	NTFS
E:	49.9 GB	50.0 GB	NTFS

The location of the file share will be a new folder in the \Shares directory on the selected volume.

○ Type a custom path:

c. Permissions page – Click Customize permissions and give the WSFC cluster object full control over the file share.



Advanced Security Settings for s1-sqlclust-2-fsw

Name: E:\Shares\s1-sqlclust-2-fsw

Owner: Administrators (S1-FILE\Administrators) Change

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

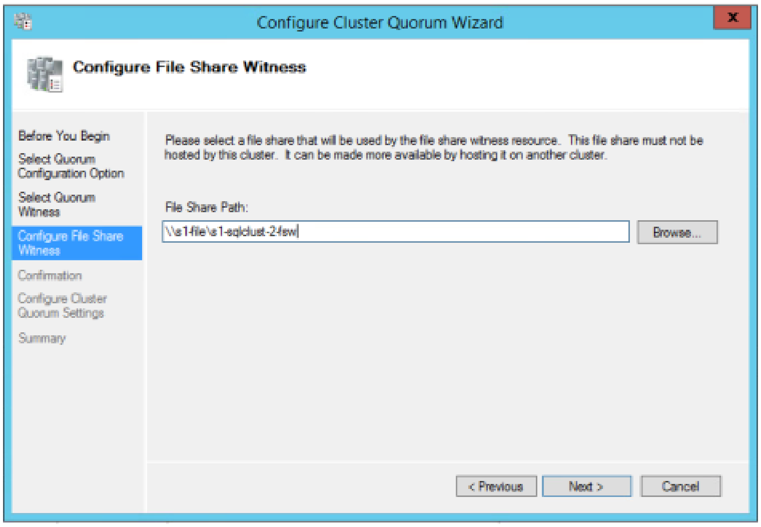
Permission entries:

Type	Principal	Access	Inherited from	Applie
Allow	Administrators (S1-FILE\Administrators)	Full control	None	This fo
Allow	Domain Admins ([redacted]\Domain Admins)	Full control	None	This fo
Allow	S1-SQLCLUST-2 ([redacted]\s1-sqlclust-2$)	Full control	None	This fo

□ Replace all child object permission entries with inheritable permission entries from this object

When you finish this step, you have two files shares: one for Site 1 and one for Site 2. For more details about accessing and using this wizard, see 12 Steps to NTFS Shared Folders in Windows Server 2012.

2. Configure the cluster quorum settings for each site:
   a. Open Failover Cluster Manager on the first VM in Site 1, right-click the cluster, and select More Actions > Configure Cluster Quorum Settings.
   b. Complete the Configure Cluster Quorum Wizard, using the following guidelines:
      i. Select Quorum Configuration Option page – Choose Select the quorum witness.
      ii. Select Quorum Witness page – Select Configure a file share witness.
      iii. Configure File Share Witness page – Enter the path to the file share you created in Step 1.


Configure Cluster Quorum Wizard
Before You Begin
Select Quorum Configuration Option
Select Quorum Witness
Configure File Share Witness
Confirmation
Configure Cluster Quorum Settings
Summary
Browse...
< Previous
Next >
Cancel

Configure File Share Witness

Please select a file share that will be used by the file share witness resource. This file share must not be hosted by this cluster. It can be made more available by hosting it on another cluster.

File Share Path: \\s1-file\s1-sqlclust-2-fsw

For detailed instructions, see Configure and manage quorum.

c. Repeat this procedure to set up a file share quorum witness in Site 2.

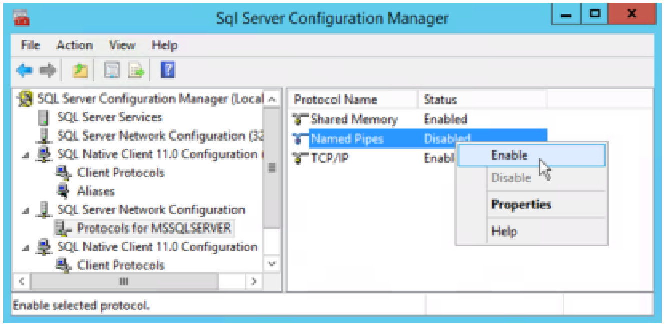
You are now ready to install App Volumes and point to the availability group we created.

Install App Volumes to use a highly available database

This procedure focuses on the specific settings required for connecting App Volumes to a highly available database. For details about other aspects of App Volumes installation, including system requirements, see the App Volumes Installation Guide.

To install App Volumes:

1. In preparation for installing App Volumes and connecting to the availability group listener, use SQL Server Configuration Manager on each SQL Server VM to enable named pipes.


Sql Server Configuration Manager
File
Action
View
Help
SQL Server Configuration Manager (Local
SQL Server Services
SQL Server Network Configuration (32
SQL Native Client 11.0 Configuration (
Client Protocols
Aliases
SQL Server Network Configuration
Protocols for MSSQLSERVER
SQL Native Client 11.0 Configuration
Client Protocols
Enable
Disable
Properties
Help

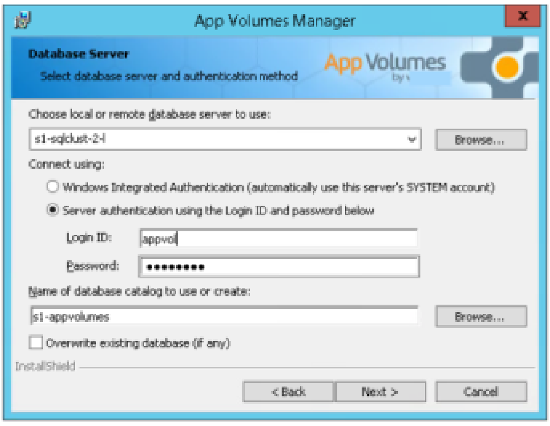
Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Disabled
TCP/IP	Enabl

Enable selected protocol.

For instructions, see Enable or Disable a Server Network Protocol.

Important: Restart the SQL Server service so the new setting can take effect.

2. In Site 1, on the first VM on which you want to install App Volumes, download the App Volumes Manager installer, start the installation wizard, and follow the prompts to the Database Server page.
3. Complete the Database Server page, using the following guidelines:
    a. Choose local or remote database server to use – Select the SQL Server Always On availability group listener that you created in Create Always On Availability Groups for App Volumes Databases.
    b. Name of database catalog to use or create – Select the database that you created in Create the App Volumes Databases and Enable Availability Groups for the Clusters.



App Volumes Manager

Database Server

Select database server and authentication method

Choose local or remote database server to use:

s1-sqlclust-2-l

Connect using:

○ Windows Integrated Authentication (automatically use this server's SYSTEM account)

● Server authentication using the Login ID and password below

Login ID: appvol

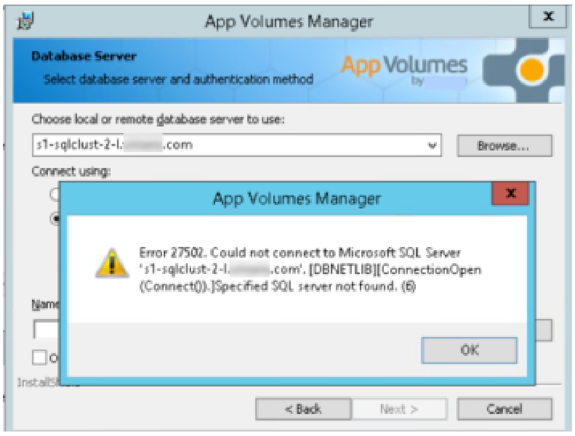
Password: ••••••••

Name of database catalog to use or create:

s1-appvolumes

□ Overwrite existing database (if any)

Important: If you see an error message such as the following one, it means you need to enable named pipes, as described in Step 1.



App Volumes Manager

Database Server

Select database server and authentication method

Choose local or remote database server to use:

s1-sqlclust-2-l.[redacted].com

Connect using:

App Volumes Manager

Error 27502. Could not connect to Microsoft SQL Server 's1-sqlclust-2-l.[redacted].com'. [DBNETLIB][ConnectionOpen (Connect()).]Specified SQL server not found. (6)

4. On the Choose Network Ports page, verify that the HTTP port is set to 80 and the HTTPS port is set to 443.
5. Follow the rest of the wizard prompts to complete the installation.
6. Repeat Steps 1–5 on the second App Volumes VM in Site 1, but on the Database Server page, be sure to leave the Overwrite existing database (if any) check box deselected.
   Both App Volumes Managers in Site 1 must point to the same highly available database.
7. Repeat this entire procedure in Site 2.

With App Volumes successfully installed, you can begin configuration. For detailed instructions see the App Volumes Administration Guide.

Multi-instance configuration

When configuring multiple instances and sites of App Volumes, you can set a source and target relationship between App Volumes instances so that packages are distributed across different App Volumes instances and locations. With App Volumes version 2203, the ability to synchronize metadata, markers, and assignments between instances was also added.

The instructions below give the process for a two-site deployment, with one instance of App Volumes per site. It can be amended to add additional sites or instances as targets to the source instance in Site 1.

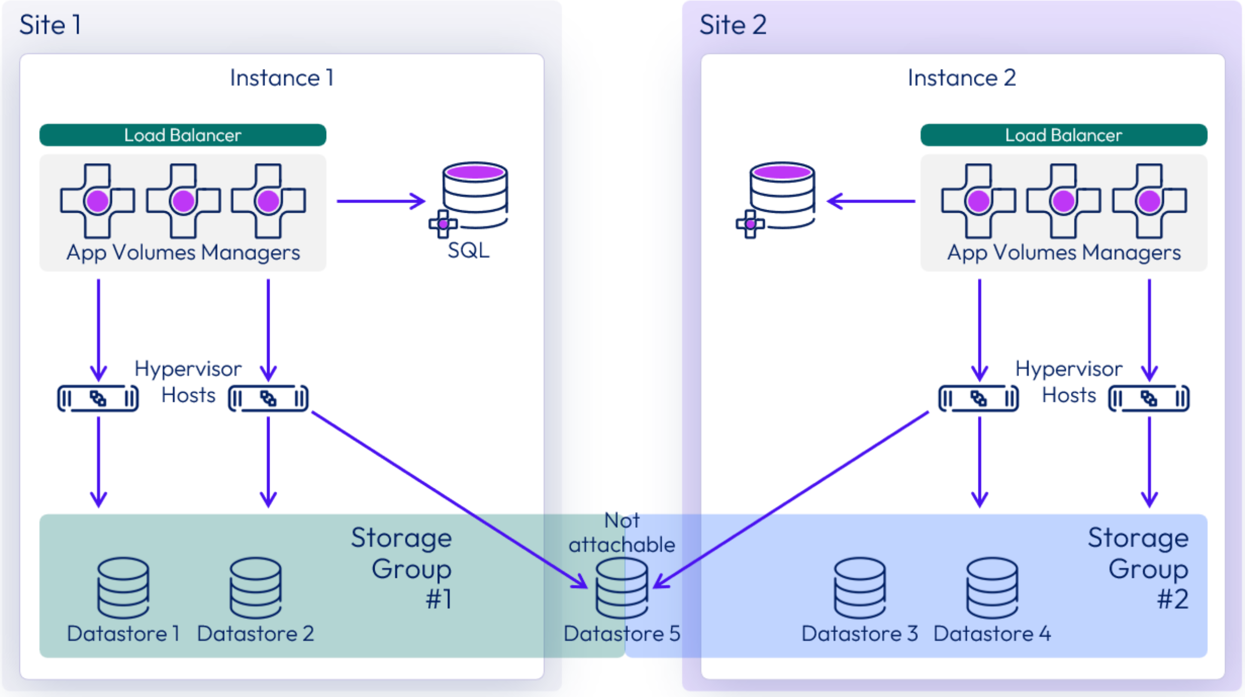


Figure 1: Example multi-instance configuration

With this design, the following is achieved:

- The Site 1 instance is defined as the source and becomes the source for packages and assignments. The instance in Site 2 is a target.
- Packages are made available in both sites.
- Packages are replicated from Site 1 (source) to Site 2 (target) through the storage groups defined, and through the use of a shared datastore that is configured as non-attachable.
- User-based entitlements for packages are replicated between sites

Install and configure App Volumes managers

1. Install App Volumes Managers:
   a. Install the first App Volumes Manager in Site 1. If using Always On availability groups to provide a local highly available database, use the local availability group listener for Site 1 when configuring the ODBC connection. Important: For step-by-step instructions on this process, see Install App Volumes to Use a Highly Available Database.
   b. Complete the App Volumes Manager wizard and add the vCenter Servers for Site 1 as machine managers, including mapping their corresponding datastores.
   c. Continue installing the subsequent App Volumes Managers for Site 1. Add them as targets to the local load balancer virtual IP.
   d. Repeat steps 1–3 for Site 2 so that the App Volumes Managers in Site 2 point to the local availability group listener for Site 2, and register the local vCenter Servers for Site 2 as machine managers.
2. Add the target instances:
   a. Using the App Volumes admin console for the Site 1 instance, add the Site 2 instance as a target instance. See Register an App Volumes Instance as a Target.
   b. Select to **enable application package import**, **synchronize markers**, and **synchronize assignments**.

Site 1 is now the source instance, and Site 2 is a target instance.

Setup package replication

This design uses a shared NFS datastore to replicate packages between sites and app volumes instances. Though NFS is not a requirement, it provides low-cost storage replication that works with both vSphere VMFS and vSAN implementations.

NFS datastores can also be used on VMC on AWS to provide a shared datastore between different SDDCs (software defined data centers).

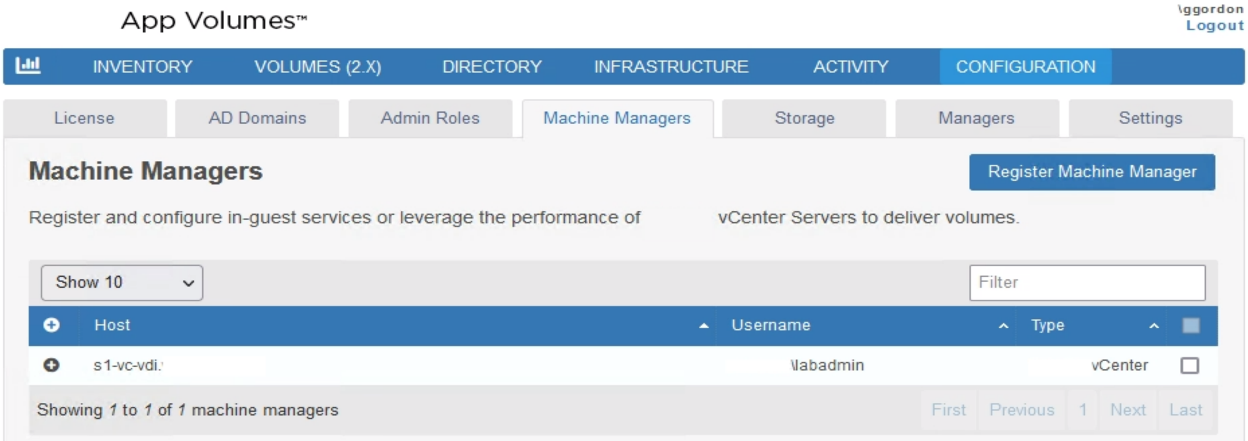
To use this setup, perform the procedures described in the sections that follow.

Configure a not attachable NFS datastore for site 1 and site 2 instances

Configure a non-attachable datastore that will act as an intermediary for package replication between the instances of App Volumes.

1. Use the App Volumes Manager console to configure a machine manager for the vCenter Server.

App Volumes–managed storage locations are available, based on the vCenter Server selected in the App Volumes machine manager configuration.



Machine Managers

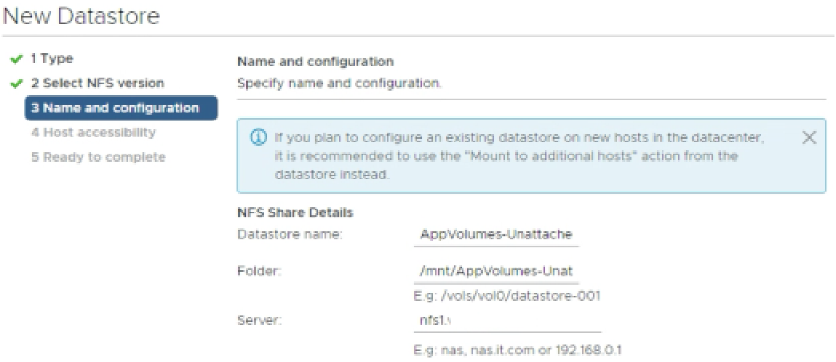
Register and configure in-guest services or leverage the performance of vCenter Servers to deliver volumes.

Host	Username	Type	
s1-vc-vdi.	\labadmin	vCenter	□

Showing *1* to *1* of *1* machine managers

2. Mount the NFS datastore to the hosts in the selected vCenter Server, to make them available as App Volumes managed storage locations.

For example, in the following screenshot, the NFS datastore AppVolumes-Unattached is added to vSphere hosts in the s1-vc-vdi.domain.com cluster.



Name and configuration

Specify name and configuration.

If you plan to configure an existing datastore on new hosts in the datacenter, it is recommended to use the "Mount to additional hosts" action from the datastore instead.

NFS Share Details

Datastore name: AppVolumes-Unattache

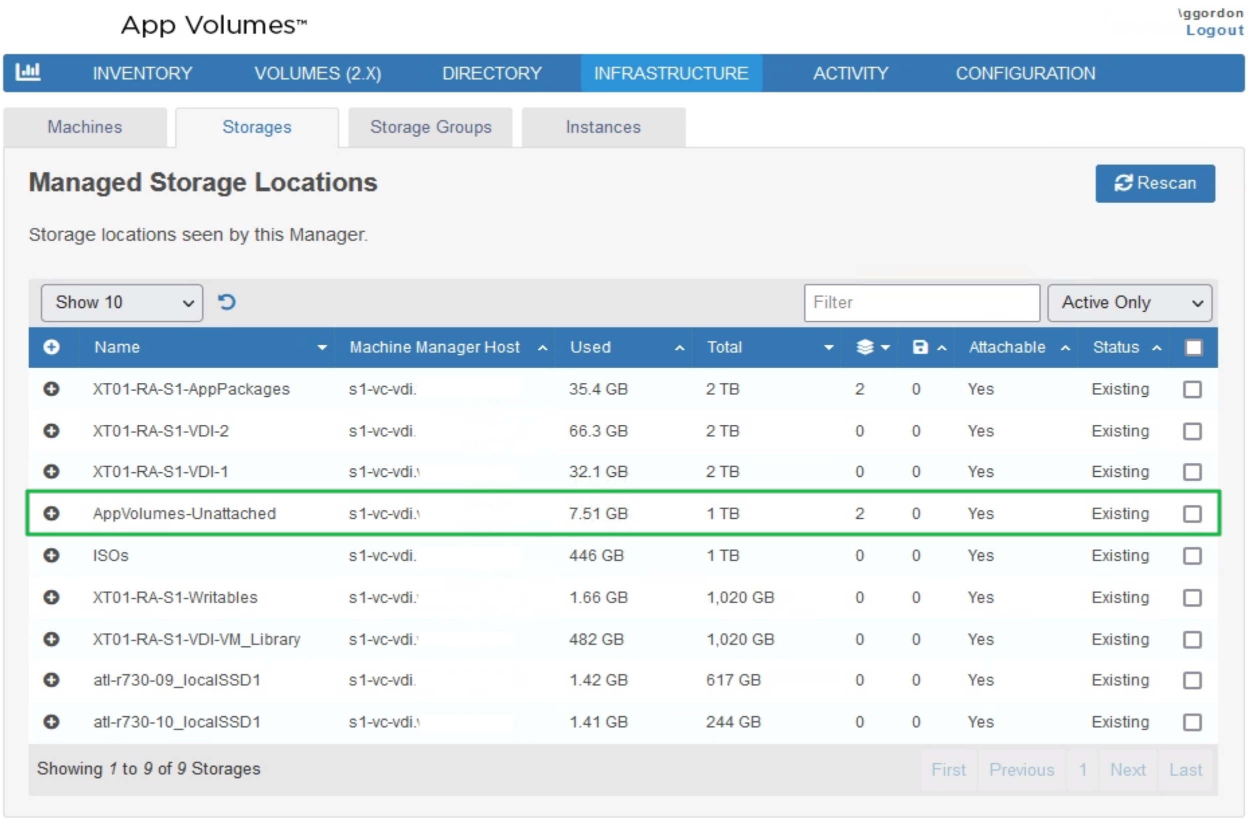
Folder: /mnt/AppVolumes-Unat

E.g: /vols/vol0/datastore-001

Server: nfs1.\

E.g: nas, nas.it.com or 192.168.0.1

3. After the NFS datastore has been added to the vSphere hosts, click Rescan to make it available as a managed storage location.



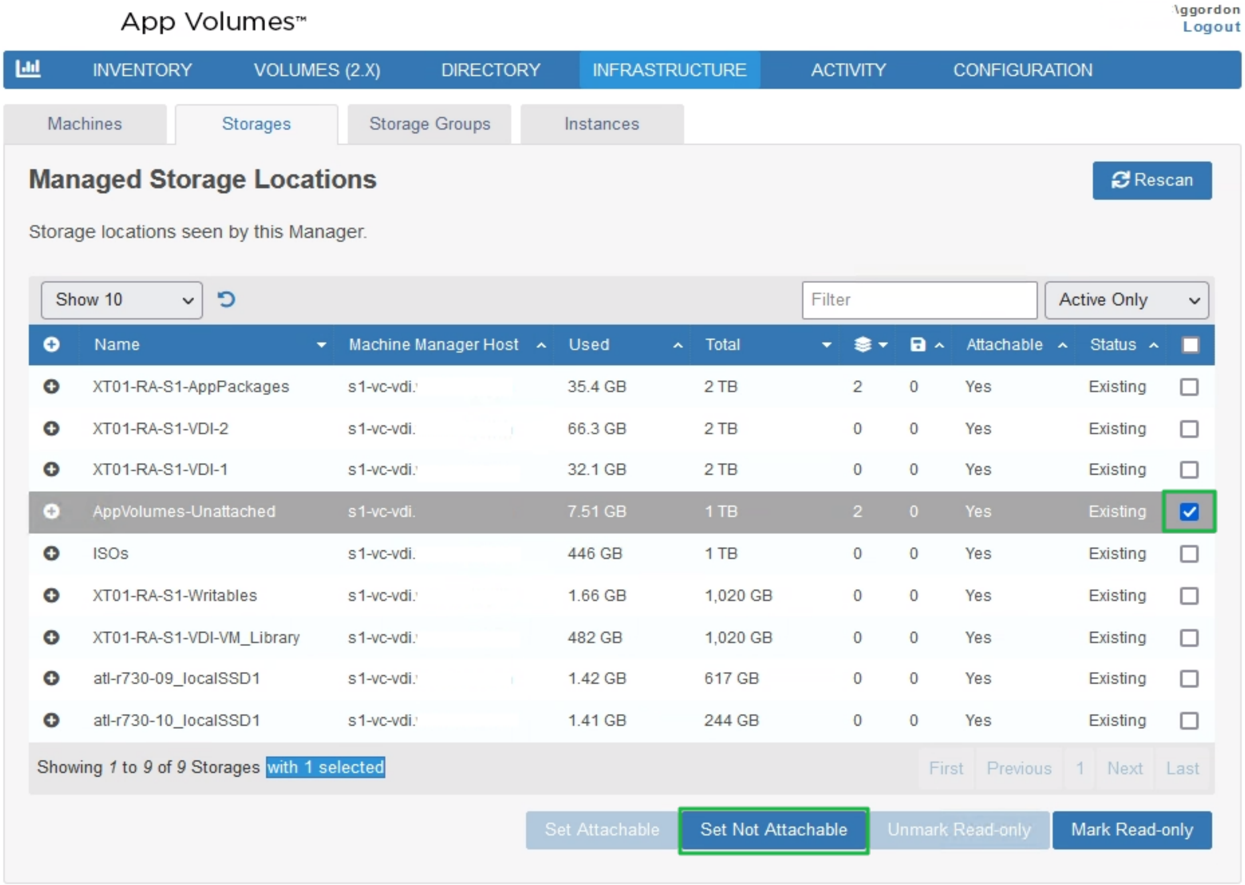
Managed Storage Locations

Storage locations seen by this Manager.

	Name	Machine Manager Host	Used	Total			Attachable	Status	□
⊕	XT01-RA-S1-AppPackages	s1-vc-vdi.	35.4 GB	2 TB	2	0	Yes	Existing	□
⊕	XT01-RA-S1-VDI-2	s1-vc-vdi.	66.3 GB	2 TB	0	0	Yes	Existing	□
⊕	XT01-RA-S1-VDI-1	s1-vc-vdi.	32.1 GB	2 TB	0	0	Yes	Existing	□
⊕	AppVolumes-Unattached	s1-vc-vdi.	7.51 GB	1 TB	2	0	Yes	Existing	□
⊕	ISOs	s1-vc-vdi.	446 GB	1 TB	0	0	Yes	Existing	□
⊕	XT01-RA-S1-Writables	s1-vc-vdi.	1.66 GB	1,020 GB	0	0	Yes	Existing	□
⊕	XT01-RA-S1-VDI-VM_Library	s1-vc-vdi.	482 GB	1,020 GB	0	0	Yes	Existing	□
⊕	atl-r730-09_localSSD1	s1-vc-vdi.	1.42 GB	617 GB	0	0	Yes	Existing	□
⊕	atl-r730-10_localSSD1	s1-vc-vdi.	1.41 GB	244 GB	0	0	Yes	Existing	□

Showing 1 to 9 of 9 Storages

4. Select the datastore, and click Set As Not Attachable. This option allows packages to replicate to and from this datastore, but prevents packages on this datastore from being mounted for use.



Managed Storage Locations

Storage locations seen by this Manager.

	Name	Machine Manager Host	Used	Total			Attachable	Status	□
⊕	XT01-RA-S1-AppPackages	s1-vc-vdi.	35.4 GB	2 TB	2	0	Yes	Existing	□
⊕	XT01-RA-S1-VDI-2	s1-vc-vdi.	66.3 GB	2 TB	0	0	Yes	Existing	□
⊕	XT01-RA-S1-VDI-1	s1-vc-vdi.	32.1 GB	2 TB	0	0	Yes	Existing	□
⊕	AppVolumes-Unattached	s1-vc-vdi.	7.51 GB	1 TB	2	0	Yes	Existing	☑
⊕	ISOs	s1-vc-vdi.	446 GB	1 TB	0	0	Yes	Existing	□
⊕	XT01-RA-S1-Writables	s1-vc-vdi.	1.66 GB	1,020 GB	0	0	Yes	Existing	□
⊕	XT01-RA-S1-VDI-VM_Library	s1-vc-vdi.	482 GB	1,020 GB	0	0	Yes	Existing	□
⊕	atl-r730-09_localSSD1	s1-vc-vdi.	1.42 GB	617 GB	0	0	Yes	Existing	□
⊕	atl-r730-10_localSSD1	s1-vc-vdi.	1.41 GB	244 GB	0	0	Yes	Existing	□

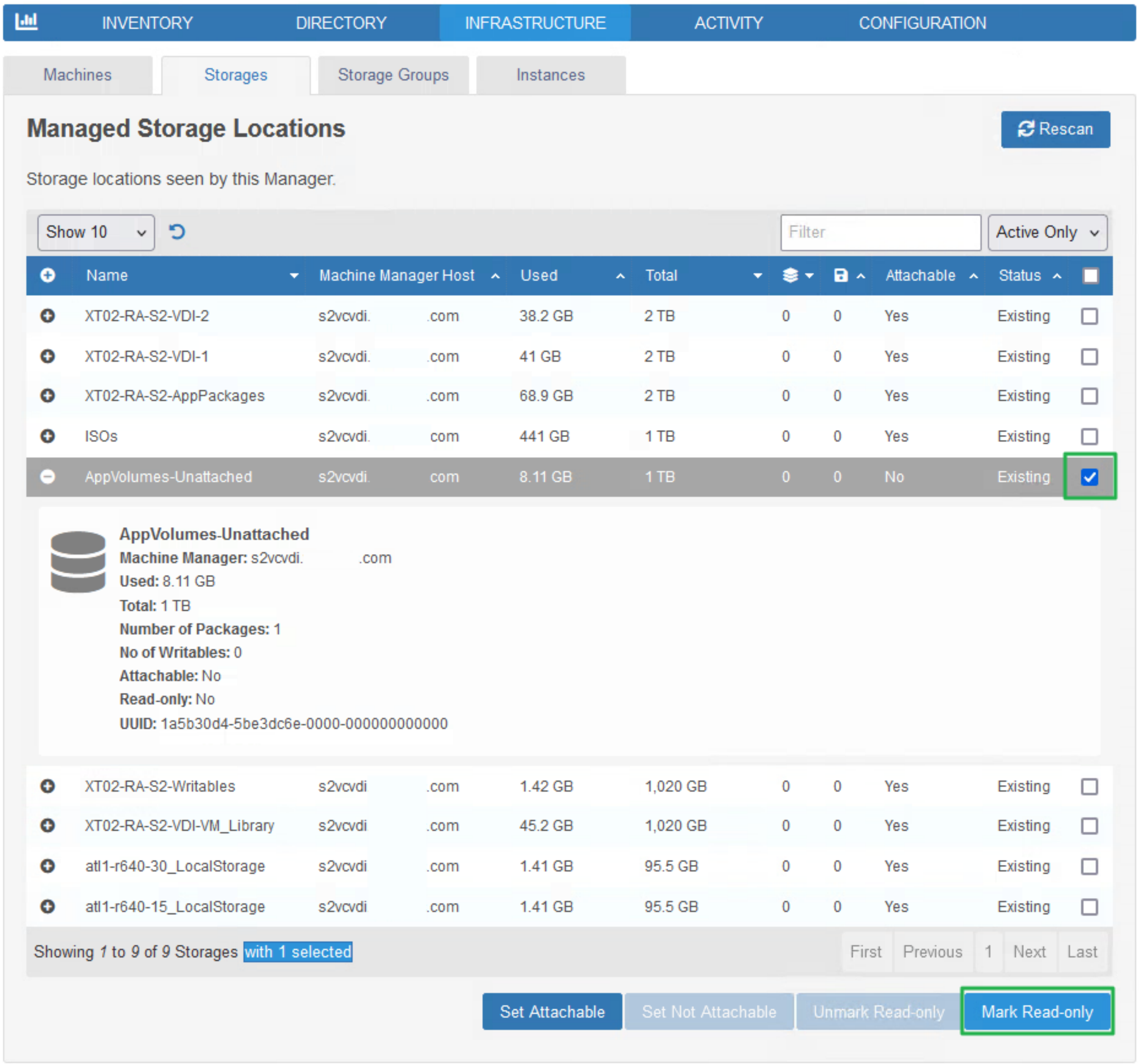
Showing 1 to 9 of 9 Storages with 1 selected

5. Repeat these steps for all other App Volumes instances.

Mark shared datastore as read-only on target instances

It is best practice to designate one instance of App Volumes as the source for package replication. This avoids possible conflicts if changes were made in two instances. On the target instances, you should mark the shared storage location, that is there for replication from the source instance, as read-only.

1. Use the App Volumes Manager console on the target App Volumes instance (Site 2 in our example).
2. Mark the shared datastore storage location as **Read-only**.



Managed Storage Locations

Storage locations seen by this Manager.

	Name	Machine Manager Host	Used	Total			Attachable	Status	☐
⊕	XT02-RA-S2-VDI-2	s2vcvdi. .com	38.2 GB	2 TB	0	0	Yes	Existing	☐
⊕	XT02-RA-S2-VDI-1	s2vcvdi. .com	41 GB	2 TB	0	0	Yes	Existing	☐
⊕	XT02-RA-S2-AppPackages	s2vcvdi. .com	68.9 GB	2 TB	0	0	Yes	Existing	☐
⊕	ISOs	s2vcvdi. com	441 GB	1 TB	0	0	Yes	Existing	☐
⊖	AppVolumes-Unattached	s2vcvdi. com	8.11 GB	1 TB	0	0	No	Existing	☑

AppVolumes-Unattached
Machine Manager: s2vcvdi. .com
Used: 8.11 GB
Total: 1 TB
Number of Packages: 1
No of Writables: 0
Attachable: No
Read-only: No
UUID: 1a5b30d4-5be3dc6e-0000-000000000000

	Name	Machine Manager Host	Used	Total			Attachable	Status	☐
⊕	XT02-RA-S2-Writables	s2vcvdi .com	1.42 GB	1,020 GB	0	0	Yes	Existing	☐
⊕	XT02-RA-S2-VDI-VM_Library	s2vcvdi .com	45.2 GB	1,020 GB	0	0	Yes	Existing	☐
⊕	atl1-r640-30_LocalStorage	s2vcvdi .com	1.41 GB	95.5 GB	0	0	Yes	Existing	☐
⊕	atl1-r640-15_LocalStorage	s2vcvdi .com	1.41 GB	95.5 GB	0	0	Yes	Existing	☐

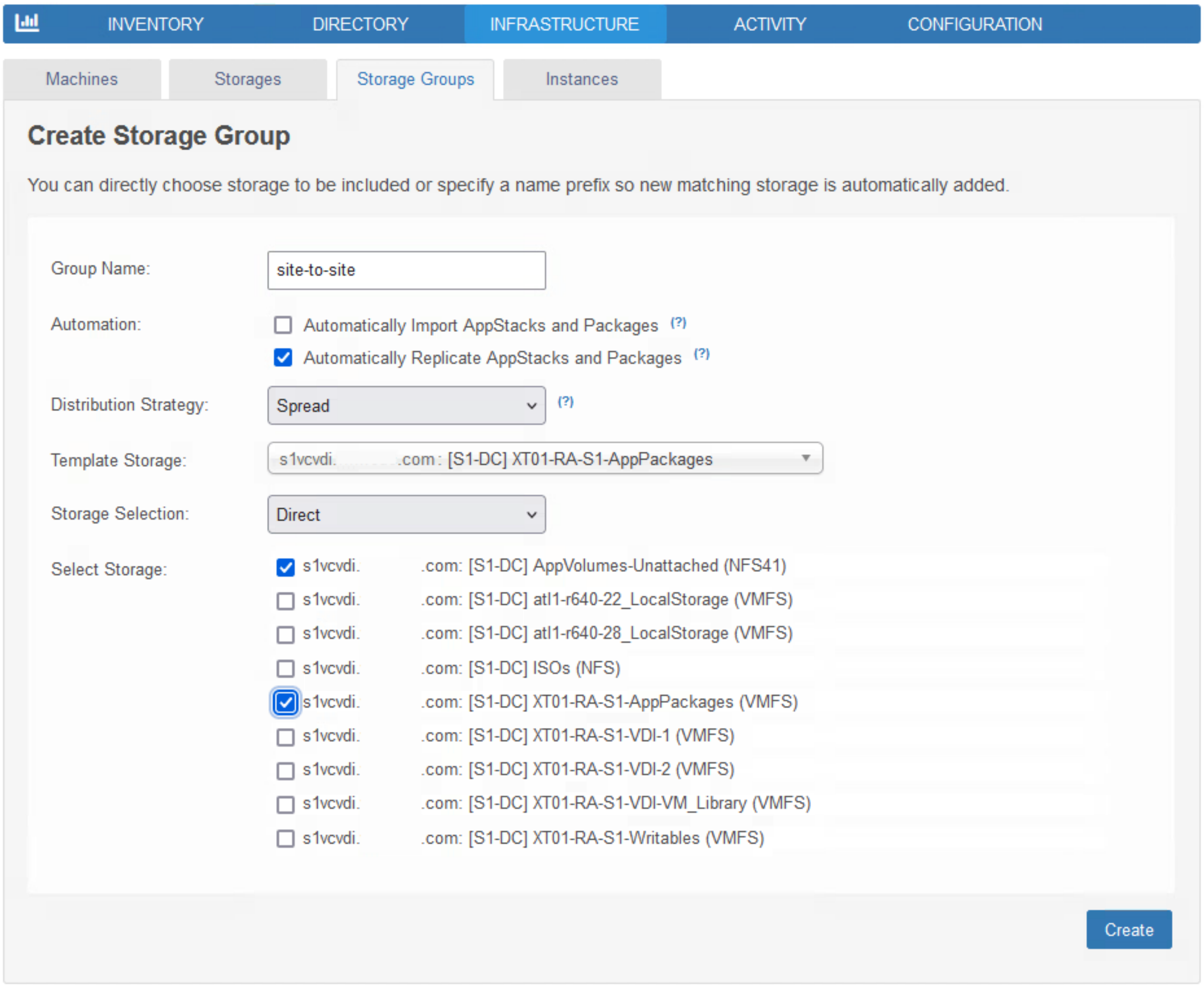
Showing *1* to *9* of *9* Storages with 1 selected

Figure 2: Mark shared datastore as read-only on target instance

Create storage groups

Next, you create a storage group that includes one or more attachable datastores, and the non-attachable datastore, as shown in the following screenshot.

1. Set up a storage group on each App Volumes instance:
   a. Create a storage group. For details, see Configure Storage Groups in App Volumes Manager.
   b. Select all datastores to be used for packages.
   c. Additionally, select one shared datastore to be used to replicate packages between instances.
   d. Do not select Automatically Import AppStacks and Packages.
   e. Select to **Automatically Replicate AppStacks and Packages**.
2. Repeat this step for the Site 2 instance, and any other target instances.



Create Storage Group

You can directly choose storage to be included or specify a name prefix so new matching storage is automatically added.

Group Name: site-to-site

Automation:
- □ Automatically Import AppStacks and Packages (?)
- ☑ Automatically Replicate AppStacks and Packages (?)

Distribution Strategy: Spread (?)

Template Storage: s1vcvdi. .com: [S1-DC] XT01-RA-S1-AppPackages

Storage Selection: Direct

Select Storage:
- ☑ s1vcvdi. .com: [S1-DC] AppVolumes-Unattached (NFS41)
- □ s1vcvdi. .com: [S1-DC] atl1-r640-22_LocalStorage (VMFS)
- □ s1vcvdi. .com: [S1-DC] atl1-r640-28_LocalStorage (VMFS)
- □ s1vcvdi. .com: [S1-DC] ISOs (NFS)
- ☑ s1vcvdi. .com: [S1-DC] XT01-RA-S1-AppPackages (VMFS)
- □ s1vcvdi. .com: [S1-DC] XT01-RA-S1-VDI-1 (VMFS)
- □ s1vcvdi. .com: [S1-DC] XT01-RA-S1-VDI-2 (VMFS)
- □ s1vcvdi. .com: [S1-DC] XT01-RA-S1-VDI-VM_Library (VMFS)
- □ s1vcvdi. .com: [S1-DC] XT01-RA-S1-Writables (VMFS)

Figure 3: Creating a storage group

Additional configuration options for Writable Volumes

You can configure writable volumes so that end users can determine how much free disk space is available in their writable volume. You can expand the amount of disk space if necessary.

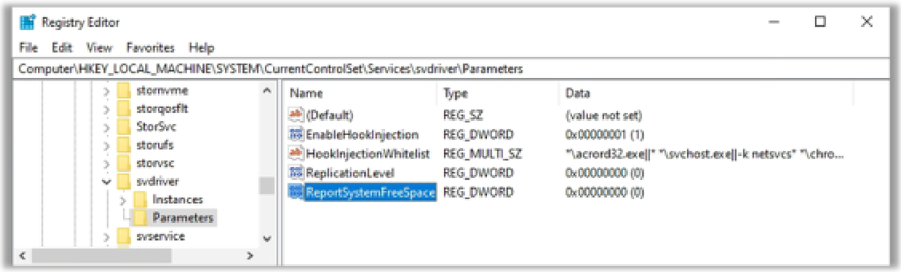
Allowing end users to see the size of Writable Volumes

Administrators can use the App Volumes Manager console to view the disk space remaining in writable volumes. Administrators can also allow end users to see how much disk space is available on their writable volume by looking at their system volume. To do this, administrators must create a new registry key during the App Volumes Agent configuration:

1. Navigate to HKLM\System\CurrentControlSet\Services\svdriver.
2. From this location, create a new registry key called Parameters if it does not already exist.
3. Within the Parameters key, create a new key called ReportSystemFreeSpace with a DWORD value of 0 (zero).

Alternatively, run a command from an elevated CMD prompt to create the correct key and value:

```
reg add hklm\system\currentcontrolset\services\svdriver\parameters /v ReportSystemFreeSpace /t REG_DWORD /d 0
```



Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\svdriver\Parameters

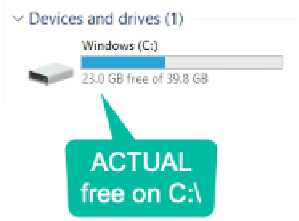
Name	Type	Data
(Default)	REG_SZ	(value not set)
EnableHookInjection	REG_DWORD	0x00000001 (1)
HookInjectionWhitelist	REG_MULTI_SZ	"\acrord32.exe||" "\svchost.exe||-k netsvcs" "\chro...
ReplicationLevel	REG_DWORD	0x00000000 (0)
ReportSystemFreeSpace	REG_DWORD	0x00000000 (0)

Figure 4: Screenshot of the Windows registry with the proper path, key, and value

Note: This change requires a reboot to take effect. Logging out or logging in does not apply the changes.

End-user viewing of Writable Volume space

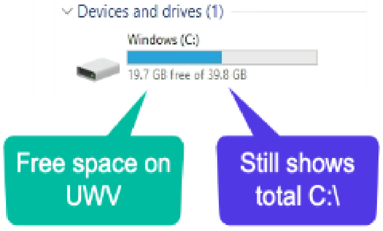
When end users look at available disk space through Windows Explorer before you make the registry changes, they can see the free space and total space reported for the C: drive.



ACTUAL free on C:\

Figure 5: Actual free space on C:\

After you make the registry modifications and reboot the system, the C: drive reports the amount of free space on the user's writable volume, which is 19.7 GB total in the following example:



Free space on UWV

Still shows total C:\

Figure 6: Free space on user Writable Volume

Notice the total space still reflects the C: drive value.

Configuring the registry so end users can view free space on their writable volumes is recommended any time you use writable volumes.

Expanding Writable Volumes

The default writable volumes template in App Volumes 4 is 10 GB. If a user’s writable volume has reached or is about to reach full capacity, it can be expanded. To expand the writable volume for each user from the App Volumes Manager console:

1. Locate the user’s writable volume on the Volumes tab under the Writables sub-tab, and expand the information on the specific writable volume.
2. Click the Expand Volume option and enter a larger value in 1-GB increments.
3. Have the user log out and log back in. The additional size is added to the writable volume after the user logs out and back in. Also, the free-space usage is not reflected in the App Volumes Manager until the user logs out and back in.

Understand packages and packaging for day 2 operations

This section builds upon the design guidance provided in the App Volumes Reference Architecture that focused on the selection of App Volumes packaging methods using App Volumes Agent and App Volumes Tools to now provide day 2 operational guidance for creating, sizing, scaling, provisioning, configuring, and updating App Volumes Packages.

Package templates

By default, a single 20-GB package template is deployed in an App Volumes environment. This template is thin-provisioned and is provided in both a VMDK and VHD format. This template can be copied and customized, depending on how large the package needs to be for a given deployment scenario.

For more information, see the Knowledge Base article Creating a new App Volumes AppStack template VMDK changing the default size of 20 GB (2116022). Note: Although this KB article was written for AppStacks, it is applicable to App Volumes 4 packages as well. With App Volumes 4.x, up to 25 packages are recommended.

If you have packages from a previous App Volumes 4 release, they will continue to work. However, additional features or fixes included in later versions are not applied to packages created with earlier versions.

Packages at scale

The number of packages that can be attached to a given VM is technically limited by the maximum number of possible drive attachments in Windows and vSphere. See the Knowledge Base article App Volumes Sizing Limits and Recommendations (67354) for guidance. In practice, the number of packages attached to a VM will likely be considerably lower than the maximum values.

Attaching packages involves the following processes:

- The disk mount (mounting the package VMDK to the VM)
- The virtualization process applied to the content in the package (merging files and registry entries with the guest OS)

The time required to complete the virtualization process varies greatly, depending on the programs contained in a given package. The more packages that need to be attached, the longer this operation might take to complete.

Packages may be assigned to a number of Active Directory objects, which has implications for the timing and specifics of which volumes are attached. See Assign an App Volumes Application to an Entity.

Recommended practices for packages in production environments

The size of the default package template is 20 GB. The default Writable Volume template is 10 GB. In some environments, it might make sense to add larger or smaller templates. For information on creating multiple, custom-sized templates, see the Knowledge Base (KB) article Creating a new App Volumes AppStack template VMDK changing the default size of 20 GB (2116022).

Note: This KB article is applicable to both App Volumes 4 packages and AppStacks. With App Volumes 4.x, up to 25 packages are recommended.

App Volumes 4 includes enhancements to the agent compared with 2.x agents. The result is improved logon times and better performance with many packages attached. Although you can likely attach more App Volumes 4 packages than App Volumes 2.x AppStacks, we recommend keeping the total number of packages assigned to a given user or computer relatively small. This can be accomplished by adding multiple programs to each package. Group the apps in such a way as to simplify distribution.

The following is a simple example for grouping App Volumes programs into packages:

- Create a package containing core programs (apps that most or all users should receive). This package can be assigned to a large group or OU.
- Create a package for departmental programs (apps limited to a department). This package can be assigned at a group or departmental level.

For traditional storage (VMFS, NFS, and so on):

- Do not place packages and VMs on the same datastore.
- Use storage groups for packages when packages are assigned to a large population of users or desktops. This helps to distribute the aggregated I/O load across multiple datastores, while keeping the assignments consistent and easy to manage.

For VMware vSAN:

- Packages and VMs can be placed on a single datastore.

- Storage groups for packages to distribute the load are not required in a vSAN implementation.
- Storage groups can still be used for the replication of packages into and out of the vSAN datastore.

See the Knowledge Base article App Volumes Sizing Limits and Recommendations (67354).

Packaging applications recommended practices

Consider the following best practices when creating and packaging applications:

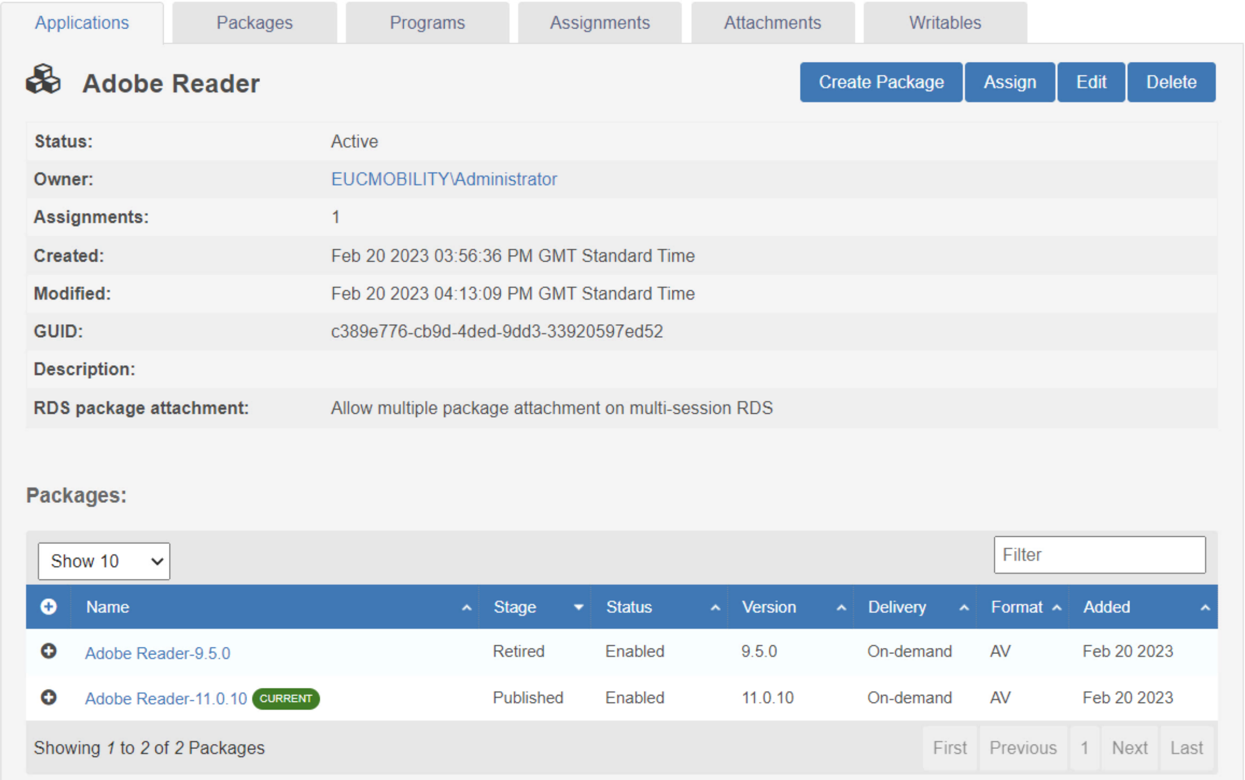
- The following characters cannot be used when naming packages: & “ ‘ < > .
- For the packaging VM, use a clean golden image that resembles as closely as possible the target environment where the package is to be deployed. For example, the packaging VM and target should be at the same OS patch and service pack level and, if programs are included in the golden image, they should also be in the packaging VM.
- The packaging VM should have the App Volumes Agent installed but should not have either the Horizon Agent or the Dynamic Environment Manager FlexEngine agent installed.
- Do not use a packaging machine where you have previously installed and then uninstalled any of the programs that you will capture. Uninstalling a program might not clean up all remnants of the software, and the subsequent App Volumes package capture might not be complete.
- Always take a snapshot of your packaging VM *before* packaging or attaching any packages to it. If any packages have been assigned to the VM, or if the VM has been used previously for packaging, revert that VM to the clean snapshot before creating a new package.

Updating and assigning updated packages recommended practices

App Volumes 4 introduced assignment types called *marker* and *package* to improve the administrative process of updating application packages. Using the CURRENT marker enables distribution of the current package to your end-user population, whereas using the package assignment type enables distribution of test versions to a subset of users for validation.

Once a new package has been tested and approved, you can simply change the CURRENT marker to point to the new package. As end users log off their desktop sessions, the old version of the package is detached. When they log on again, the new version is attached.

The following illustration shows the Adobe Reader application, which contains two packages with different versions of the software. Adobe Reader-11.0.10 has the CURRENT marker, so it is used for any user who has an assignment to the Adobe Reader application, using a marker assignment type, with the current marker.



Adobe Reader

Status:	Active
Owner:	EUCMOBILITY\Administrator
Assignments:	1
Created:	Feb 20 2023 03:56:36 PM GMT Standard Time
Modified:	Feb 20 2023 04:13:09 PM GMT Standard Time
GUID:	c389e776-cb9d-4ded-9dd3-33920597ed52
Description:	
RDS package attachment:	Allow multiple package attachment on multi-session RDS

Packages:

Show 10 | Filter

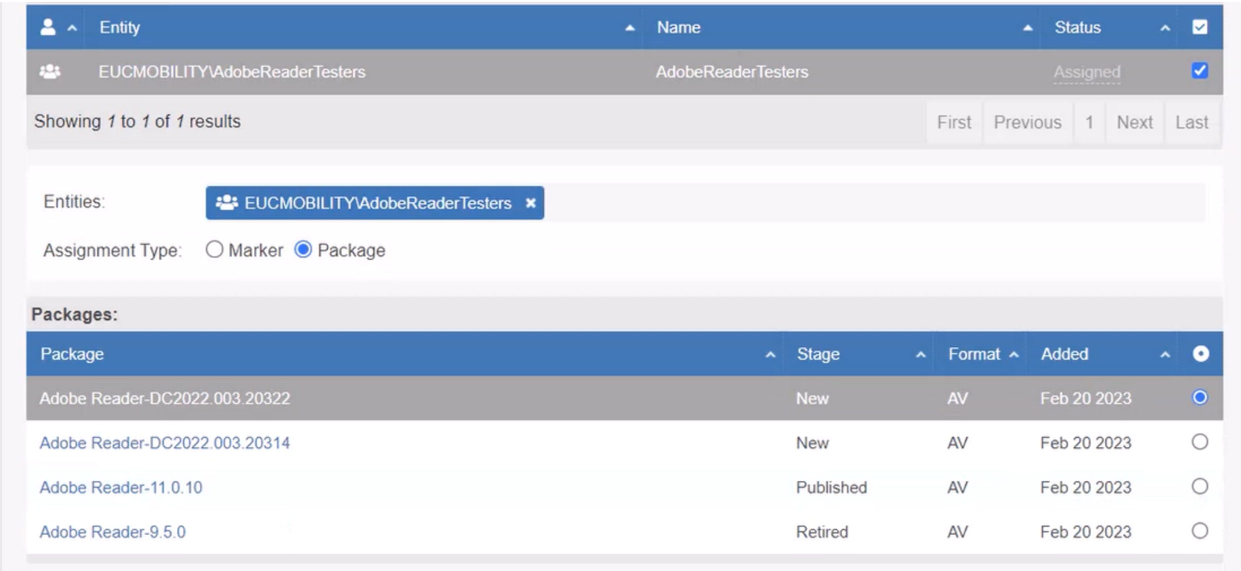
⊕	Name	Stage	Status	Version	Delivery	Format	Added
⊕	Adobe Reader-9.5.0	Retired	Enabled	9.5.0	On-demand	AV	Feb 20 2023
⊕	Adobe Reader-11.0.10 CURRENT	Published	Enabled	11.0.10	On-demand	AV	Feb 20 2023

Showing *1* to *2* of *2* Packages

Figure 10: Application with two package versions with current marker

When new version of an application is to be rolled out to users, a new package version can be captured, in this case Adobe Reader-DC2022.003.20322. You can either update an existing package version or create a new package from scratch. When updating an existing package, this clones the original package for you to work with and apply updates. End users continue to work from the original package to prevent user downtime.

Before pushing the new version of this out to users, it is best practice to assign it to a group of test users to confirm functionality. This can be achieved by creating an assignment for those test users that uses the Package assignment type.



	Entity	Name	Status	☑
	EUCMOBILITY\AdobeReaderTesters	AdobeReaderTesters	Assigned	☑

Showing *1* to *1* of *1* results

Entities: EUCMOBILITY\AdobeReaderTesters ✖

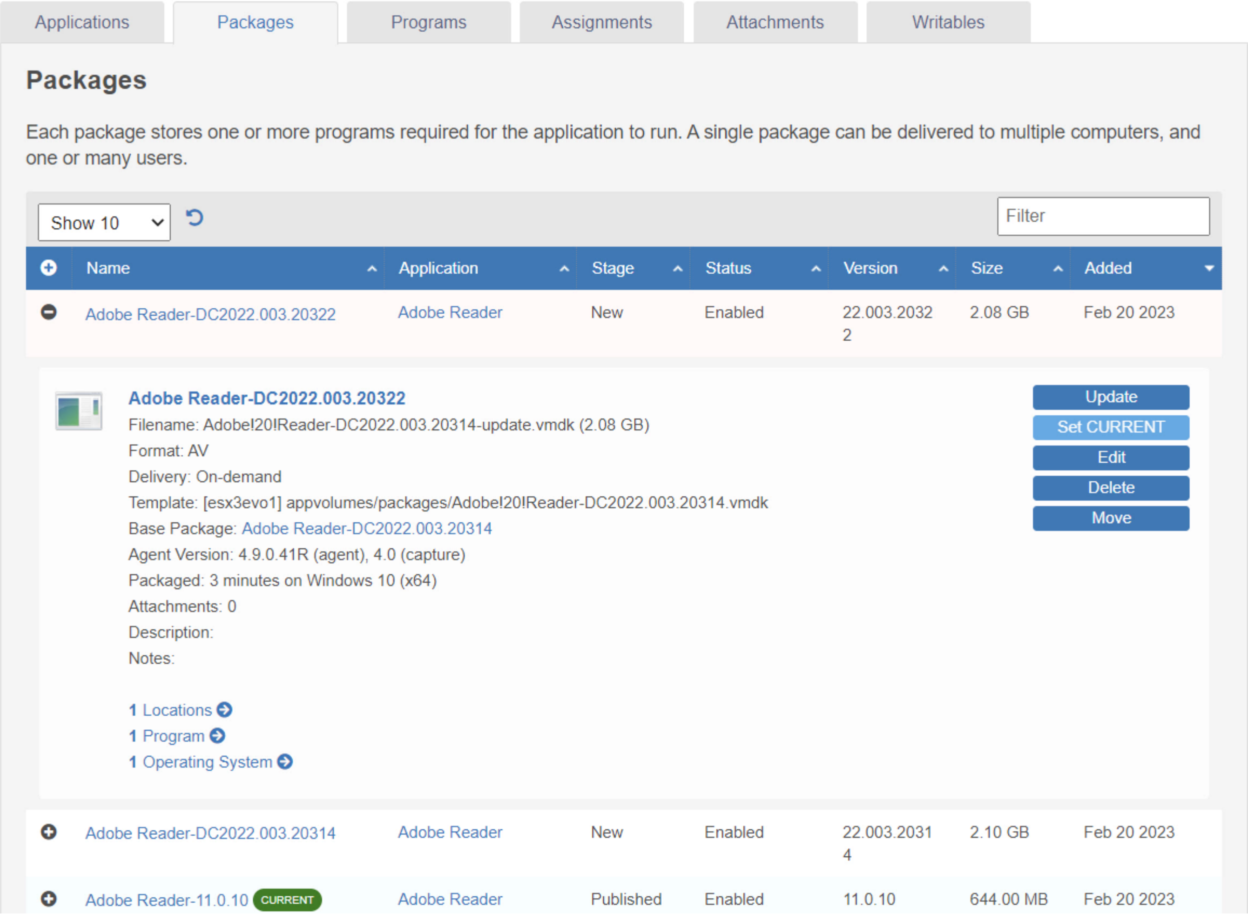
Assignment Type: ○ Marker ● Package

Packages:

Package	Stage	Format	Added	
Adobe Reader-DC2022.003.20322	New	AV	Feb 20 2023	●
Adobe Reader-DC2022.003.20314	New	AV	Feb 20 2023	○
Adobe Reader-11.0.10	Published	AV	Feb 20 2023	○
Adobe Reader-9.5.0	Retired	AV	Feb 20 2023	○

Figure 11: Using a package marker assignment

Once tested, you can simply swap the CURRENT marker to the new Package version that you want users to get. In the example below, we set the CURRENT marker to the Adobe Reader-DC2022.003.20322 package. Any user who has an assignment to Adobe Reader application with the CURRENT marker type will now get the new version.



Packages

Each package stores one or more programs required for the application to run. A single package can be delivered to multiple computers, and one or many users.

Name	Application	Stage	Status	Version	Size	Added
Adobe Reader-DC2022.003.20322	Adobe Reader	New	Enabled	22.003.20322	2.08 GB	Feb 20 2023
Adobe Reader-DC2022.003.20314	Adobe Reader	New	Enabled	22.003.20314	2.10 GB	Feb 20 2023
Adobe Reader-11.0.10 CURRENT	Adobe Reader	Published	Enabled	11.0.10	644.00 MB	Feb 20 2023

Adobe Reader-DC2022.003.20322
Filename: Adobe!20!Reader-DC2022.003.20314-update.vmdk (2.08 GB)
Format: AV
Delivery: On-demand
Template: [esx3evo1] appvolumes/packages/Adobe!20!Reader-DC2022.003.20314.vmdk
Base Package: Adobe Reader-DC2022.003.20314
Agent Version: 4.9.0.41R (agent), 4.0 (capture)
Packaged: 3 minutes on Windows 10 (x64)
Attachments: 0
Description:
Notes:

1 Locations
1 Program
1 Operating System

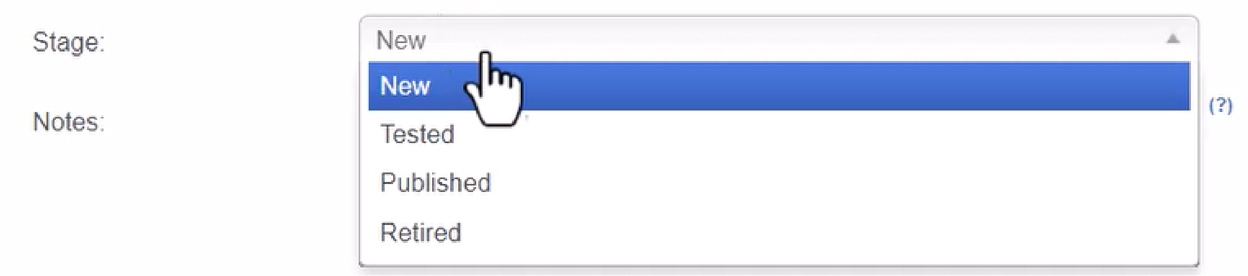
Figure 12: Setting the CURRENT marker to a new package version

To learn more about assignment types refer to *Assign Application Package* in App Volumes 4 Feature Review.

To initiate an update to programs in an existing package, use the App Volumes Manager Admin UI to invoke the update process. This process clones the original package for you to work with and apply updates. End users continue to work from the original package to prevent user downtime. The new package with the updated programs is distributed by simply moving the CURRENT marker once it has been approved.

Consider the following best practices when updating and assigning updated packages:

- When creating and updating packages, use the Stage drop-down list to select an appropriate value. This makes it easy for you and other App Volumes admins to manage the lifecycle of the package.


Stage: New

Notes:

Figure 13: Define the stage for a package

- Use the marker assignment type to simplify updates for your general population of users.
- Use the Unset CURRENT option to prevent delivery of a package without modifying assignments.
- Use the package assignment type for one-off, explicit assignments of a specific version.

Note**:** If both package and marker assignments are made, the package assignment is used.

Performance testing for packages

Test packages immediately after packaging to determine their overall performance. Using a performance analytics tool, gather virtual machine, host, network, and storage performance information for use when packages are operated on a larger scale. Do not neglect user feedback, which can be extremely useful for assessing the overall performance of an application.

Because App Volumes provides an application container and brokerage service, storage performance is very important in a production environment. Packages are read-only. Depending on utilization patterns, the underlying shared storage platform might have significant read I/O activity. Consider using flash and hybrid-flash storage technologies for packages.

This evaluation can be time-consuming for the administrator, but it is necessary for any desktop- transformation technology or initiative.

ThinApp integration with packages

Network latency is often the limiting factor for scalability and performance when deploying ThinApp packages in streaming mode. But ThinApp provides exceptional application-isolation capabilities. With App Volumes, administrators can present ThinApp packages as dynamically attached applications that are located on storage rather than as bits that must traverse the data center over the network.

Using App Volumes to deliver ThinApp packages removes network latency due to Windows OS and environmental conditions. It also allows for the best of both worlds—real-time delivery of isolated and troublesome applications alongside other applications delivered on packages.

With App Volumes in a virtual desktop infrastructure, enterprises can take advantage of local deployment mode for ThinApp packages. ThinApp virtual applications can be provisioned inside a package using all the storage options available for use with packages. This architecture permits thousands of virtual desktops to share a common ThinApp package through packages without the need to stream or copy the package locally.

Microsoft Office application packages

For guidance on deploying Microsoft Office in a Horizon environment, see Best Practices for Delivering Microsoft Office 365 in Horizon.

Core Microsoft Office applications can either be packaged with App Volumes or included in the base golden image used to create the desktop pool. Factors to consider include your preference in application packaging, and whether all users will get access to core Microsoft Office applications. See Application Suitability for Packages to understand some of the considerations.

Non-core Microsoft Office applications, such as Visio and Project, can be packaged separately with App Volumes and then assigned to only the users that require those applications. Versions of core and non-core Office that are delivered to the same VM, should be the same version.

Note: Ensure that core and non-core Microsoft Office applications are of similar version and patch levels.

Office plug-ins and add-ons

The most straightforward method is to include Microsoft Office plug-ins or add-ons in the same package as the Microsoft Office installation.

However, if necessary, you can include plug-ins or add-ons in packages that are separate from the packages that contain the Microsoft applications to which they apply. Before packaging the plug-in or add-on, install the primary application natively in the OS of the packaging VM.

Note: Ensure the plug-in or add-on is at the same version as the Microsoft Office application in the package. This includes any patches or updates.

RDSH integration with packages

App Volumes supports package integration with Microsoft RDSH-published desktops and published applications. Packages are assigned to RDSH servers rather than directly to users. Users are then entitled to the RDSH-published desktops or applications through the Horizon entitlement process.

- When using Horizon Published Apps on Demand, App Volumes packages are attached to the RDSH server when a user launches the application.
- Otherwise, packages are attached to the RDSH server when the machine is powered on, and the App Volumes service starts.

Note: Writable Volumes are not supported with RDSH assignments.

Published Apps on Demand is recommended as it gives flexibility and simplifies the farm operations. If you are not using Apps on Demand but instead assigning packages to specific RDSH farms:

- Consider associating packages at the OU level in Active Directory, rather than to individual computer objects. This practice reduces the number of package entitlements and ensures packages are always available as new hosts are created and existing hosts are refreshed.
- Entitling packages to an OU where Horizon instant-clone RDSH server farms are provisioned ensures that all hosts are configured exactly alike, which supports dynamic growth of farms with minimal administrative effort.

Create dedicated packages for RDSH servers. Do not reuse a package that was originally created for a desktop OS.

When creating the package, install programs on a packaging machine that has the same operating system as that used on the deployed RDSH servers. Before installing software, switch the RDSH server to RD-Install mode. For more information, see Learn How To Install Applications on an RD Session Host Server.

See System Requirements to verify that the Windows Server version you want to use for RDSH is supported for the App Volumes Agent.

For information about using App Volumes in a Citrix Virtual Apps and Desktops shared-application environment, see App Volumes Deployment Guide for Citrix Virtual Apps and Desktops.

Application suitability for packages

Most Windows applications work well with App Volumes, including those with services and drivers, and require little to no additional interaction. If you need an application to continue to run after the user logs out, it is best to natively install this application on the desktop or desktop image.

The following sections briefly describe situations and application types where App Volumes might need special attention to work properly or where the application would work best when installed in the golden image, rather than in a package.

Applications that work best in the golden image

Applications that should be available to the OS in the event that a package or Writable Volume is not present should remain in the golden image and not in an App Volumes virtual disk. These types of applications include antivirus, Windows updates, and OS and product activations, among others. Applications that should be available to the OS when no user is logged in should also be placed in the golden image.

Similarly, applications that integrate tightly with the OS should not be virtualized in a package. If these apps are removed from the OS in real time, they can cause issues with the OS. Again, if the application needs to be present when the user is logged out, it must be in the golden image and not in a package. Applications that start at boot time or need to perform an action before a user is completely logged in, such as firewalls, antivirus, and Microsoft Internet Explorer, fall into this category.

Applications that use the user profile as part of the application installation should not be virtualized in an App Stack. App Volumes does not capture the user profile space C:\users\<username>. If, as part of its installation process, an application places components into this space, those components will not be recorded as part of the packaging process. If this happens, undesired consequences or failure of the application might result when the application is captured in a package.

Applications whose components are not well understood

In the rare event that an issue with an application does present itself, it is important to have a thorough understanding of how the application functions. Understanding the processes that are spawned, file and registry interactions, and where files are created and stored is useful troubleshooting information.

App Volumes is a delivery mechanism for applications. It is important to understand that App Volumes does an intelligent recording of an installation during the packaging process and then delivers that installation. If the installation is not accurate or is configured incorrectly, the delivery of that application will also be incorrect (“garbage in, garbage out”). It is important to verify and test the installation process to ensure a consistent and reliable App Volumes delivery.

App Volumes Agent altitude and interaction with other mini-filter drivers

The App Volumes Agent is a mini-filter driver. Microsoft applies *altitudes* to filter drivers. The concept is that the larger the number, the “higher” the altitude. Mini-filter drivers can see only the other filter drivers that are at a higher altitude. The actions at a lower altitude are not seen by filter drivers operating at a higher altitude.

The lower-altitude mini-filter drivers are the first to interact with a request from the OS or other applications. Generally speaking, the requests are then given to the next mini-filter driver in the stack (next highest number) after the first driver finishes processing the request. However, this is not always the case because some mini-filter drivers might not release the request and instead “close” it out to the OS or application.

In the case where a request is closed, the subsequent mini-filter drivers will never see the request at all. If this happens with an application running at a lower altitude than App Volumes, the App Volumes mini-filter driver will never get a chance to process the request, and so will not be able to virtualize the I/O as expected.

This is the primary reason that certain applications that use a mini-filter driver should be turned off or removed from the OS while you install applications with App Volumes. There might be additional scenarios where App Volumes Agent should be turned off, allowing other applications to install correctly in the base OS.

Other special considerations

The following guidelines will also help you determine whether an application requires special handling during the virtualization process or whether virtualization in a package is even possible:

- Additional application virtualization technologies – Other application virtualization technologies (Microsoft App-V, ThinApp, and others) should be turned off during packaging because the filter drivers could potentially conflict and cause inconsistent results in the packaging process.
- Mixing of 32- and 64-bit OS types – The OS type (32- or 64-bit) of the machine that the package is attached to should match the OS type that applications were packaged on. Mixing of application types in App Volumes environments follows the same rules as Windows application types—that is, if a 32-bit application is certified to run in a 64-bit environment, then App Volumes supports that configuration also.
- Exceptional applications – Some software apps just do not work when installed on an App Volumes package. There is no list of such applications, but an administrator might discover an issue where an application simply does not work with App Volumes.

In summary, most applications work well with App Volumes, with little to no additional interaction needed. However, you can save time and effort by identifying potential problems early, by looking at the application type and use case before deciding to create a package.

Recommended practices for production environments

The following recommendations pertain to golden image optimization, setting up virtual desktops, and security best practices.

Golden image optimization

Golden images should be optimized for VDI or RDSH to ensure the best performance possible in a virtualized environment. Consider using the instructions in Manually Creating Optimized Windows Images for Horizon VMs when building your golden images. The Windows OS Optimization Tool for Horizon helps optimize Windows desktop and server operating systems for use with Horizon.

The OS Optimization Tool includes a customizable template to enable or turn off Windows system services and features across multiple systems, per recommendations and best practices. Because most Windows system services are enabled by default, the OS Optimization Tool can be used to easily turn off unnecessary services and features to improve performance.

Desktop pool recommendations

When setting up client endpoint devices, consider the following best practices:

- When reverting a desktop VM that is running the App Volumes Agent to a previous snapshot, make sure that the VM is gracefully shut down, to avoid synchronization issues. This is primarily relevant to the packaging desktop and golden image VMs for Horizon linked- and instant-clone desktop pools.
- If you are using a Horizon desktop pool, the App Volumes Agent should be installed on the golden image VM for linked- and instant-clone pools, or on the VM template for full-clone pools, for ease of distribution.
- If using a Horizon linked-clone pool, make sure the Delete or Refresh machine on logoff policy in Desktop Pool Settings is set to Refresh Immediately. This policy ensures that the VMs stay consistent across logins.

Summary and additional resources

Now that you have come to the end of this configuration chapter on Omnissa App Volumes, you can return to the reference architecture landing page and use the tabs, search, or scroll to select further chapter in one of the following sections:

- **Overview chapters** provide understanding of business drivers, use cases, and service definitions.
- **Architecture chapters** give design guidance on the Omnissa products you are interested in including in your deployment, including Workspace ONE UEM, Access, Intelligence, Workspace ONE Assist, Horizon Cloud Service, Horizon 8, App Volumes, Dynamic Environment Manager, and Unified Access Gateway.
- **Integration chapters** cover the integration of products, components, and services you need to create the environment capable of delivering the services that you want to deliver to your users.
- **Configuration chapters** provide reference for specific tasks as you deploy your environment, such as installation, deployment, and configuration processes for Omnissa Workspace ONE, Horizon Cloud Service, Horizon 8, App Volumes, Dynamic Environment Management, and more.

Additional resources

For more information about App Volumes, you can explore the following resources:

- App Volumes product page
- App Volumes documentation
- Knowledge Base

Changelog

The following updates were made to this guide:

Date	Description of Changes
2025-02-15	
2024-05-27	
2023-07-25	
2022-05-25	
2020-07-01	

Author and contributors

This chapter was written by:

- Graeme Gordon, Senior Staff Architect, Omnissa.
- Donal Geary, Senior Lab Manager, Omnissa.
- Josh Spencer, Senior Product Line Manager, Omnissa.

Feedback

Your feedback is valuable. To comment on this paper, either use the feedback button or contact us at tech_content_feedback@omnissa.com.