APP VOLUMES DEPLOYMENT CONSIDERATIONS
VMware App Volumes 2.12 and Later
Table of Contents

Introduction

App Volumes Overview

– JMP – Next-Generation Desktop and Application Delivery Platform

How App Volumes Works

Summary of App Volumes Benefits

App Volumes Components and Architecture

– App Volumes Components
– Architecture of an App Volumes Deployment
– Data Flow in an App Volumes Environment

Managing Applications with AppStacks

– AppStack Design
– Application Suitability
– ThinApp Integration
– RDSH Integration
– Microsoft Office Applications on AppStacks
– AppStack Templates

App Volumes Writable Volumes

– Writable Volume Templates
– Expandable Writable Volumes
– Allowing End Users to See the Size of Writable Volumes
End-User Viewing of Writable Volume Space

Expanding Writable Volumes

Writable Volume Policies

Storage Groups

AppStack Storage Groups

Non-Attachable AppStack Datastores

Writable Volume Storage Groups

Deployment Requirements

Planning for Production

Storage Planning

AppStack and Writable Volume Template Placement

Free-Space Considerations

Writable Volumes Delay Creation Option

Mount on Host

Mount Local

Performance

Scalability

View and vSphere at Scale

Multi-vCenter-Server Considerations

Multiple Active Directory Considerations

App Volumes SQL Database at Scale

App Volumes Manager Servers at Scale

AppStacks at Scale
Security
Availability
Recoverability

Deployment Procedures

- Install App Volumes Manager
- Configure App Volumes Manager
- Install App Volumes Agent
- Build a Provisioning Virtual Machine
- Prepare the Provisioning Machine
- Install the App Volumes Agent on the Provisioning Machine
- Take a Virtual Machine Snapshot
- Create AppStack and Writable Volume
- AppStack Creation and Assignment
- Writable Volume Creation and Assignment

Best Practices for an App Volumes Deployment in Production

- Database Best Practices
- Master Image Best Practices
- Best Practices for Creating and Provisioning AppStacks
- Best Practices for Configuring AppStacks
- Best Practices for Updating AppStacks and Assigning Updated AppStacks
- Sizing Best Practices for AppStacks
- App Volumes Manager Storage Policy Best Practices
- Best Practices for Client Desktops
– Miscellaneous Best Practices

Additional Resources

About the Authors and Contributors

– Contributors
App Volumes Deployment Considerations

Introduction
This deployment considerations guide describes VMware App Volumes™ capabilities, architecture, and implementation requirements and addresses frequently asked high-level questions about deploying an App Volumes solution. The example setting for this deployment is a View virtual desktop environment in VMware Horizon® 7.

App Volumes Overview
App Volumes is a real-time application-delivery and life-cycle-management tool.

Figure 1: App Volumes Application Management Containers

Enterprises can use App Volumes to build real-time application-delivery systems that ensure applications are centrally managed. Applications are delivered to desktops through virtual disks. With the App Volumes solution, you do not need to modify desktops or applications themselves, and you can scale out easily and cost-effectively, without compromising end-user experience.

App Volumes is an important part of the VMware Horizon 7 Enterprise Edition, which provides an application-delivery mechanism for VDI desktops as well as RDSH-hosted desktops and applications.
JMP – Next-Generation Desktop and Application Delivery Platform

JMP (pronounced jump) represents capabilities in VMware Horizon 7 Enterprise Edition that deliver Just-in-Time Desktops and Apps in a flexible, fast, and personalized manner. JMP is composed of the following VMware technologies:

- VMware Instant Clone Technology for fast desktop and RDSH provisioning
- VMware App Volumes for real-time application delivery
- VMware User Environment Manager™ for contextual policy management

JMP allows components of a desktop or RDSH server to be decoupled and managed independently in a centralized manner, yet reconstituted on demand to deliver a personalized user workspace when needed. JMP is supported with both on-premises and cloud-based Horizon 7 deployments, providing a unified and consistent management platform regardless of your deployment topology. The JMP approach provides several key benefits, including simplified desktop and RDSH image management, faster delivery and maintenance of applications, and elimination of the need to manage “full persistent” desktops.

How App Volumes Works

In modern desktop environments, the demand for real-time application delivery often puts strain on existing infrastructures. Through App Volumes, VMware addresses this strain by virtualizing applications above the operating system (OS) and by offering an alternative to managing per virtual machine.

Applications virtualized above the OS are delivered as if natively installed—without modification, in various configurations, to multiple groups of users. And, through file and registry virtualization, applications are organized into application management containers. This arrangement uses existing storage and networking services, reduces infrastructure strain and overhead, and simplifies application life-cycle management.

As illustrated in Figure 1, application-management containers are above the desktop OS, which has an App Volumes Agent installed. Applications, data files, middleware, and configurations are in separate, layered containers. There are two types of App Volumes containers:

- **AppStack** – A read-only container for one-to-many delivery of IT-managed applications
- **Writable volume** – A one-to-one, user-specific, read-and-write container for user-installed applications, data, and files

Administrators use the App Volumes Manager, a Web-based interface integrated with Active Directory (AD) and VMware vSphere®, to create AppStacks and assign application entitlements. Application installations recorded in AppStacks are stored in shared volumes across virtual disks. These applications require no special packaging formats or snapshot technologies and are easy to provision. During provisioning of AppStacks, App Volumes intelligently records the entire application-installation process, and the changes made by the native application installers are available for delivery to users.

Administrators can easily deliver provisioned AppStacks to an individual system, a user, or a group. Applications delivered by App Volumes look and feel natively installed, and they follow users across sessions and devices, as can data, at the administrator’s option. Administrators can update or replace applications in real time and remove any assigned application, either immediately, while the user is still logged in, or, in accordance with VMware best practices, at next login or reboot.

With a writable volume, user data, such as user-installed applications, moves across systems with the user. Any settings that the user applies to an application are stored in the writable volume—these can be settings for an application either on an AppStack or on the user’s writable volume. **VMware User Environment Manager** complements the App Volumes AppStack feature. User Environment Manager can manage application settings for all users of a particular AppStack at a more granular level, and provide contextual rules to enforce policy, based on different conditions or events. User Environment Manager can also manage application settings for writable volumes assigned to single users.

Summary of App Volumes Benefits

With App Volumes, applications become objects that can be moved easily across data centers or to the cloud and shared with thousands of virtual machines. In a virtual desktop environment, App Volumes provides the following benefits:

- Real-time, dynamic application delivery in virtualized environments
Delivers and updates applications and data by using a one-to-many shared delivery model.
Applications delivered through App Volumes behave as natively installed applications.

Persistent end-user experience in nonpersistent environments
- Supports fully customizable desktops, allowing end users to install their own applications on writable volumes.
- Invisibly attaches applications to virtual machines, undetected by end users.
- Creates a persistent user experience with the cost savings of a nonpersistent architecture.

Application life-cycle management
- Manages the application life cycle, from provisioning, delivery, and maintenance, to retirement.
- Speeds up application updates or upgrades and supports easy application replacement.

Reduced VDI infrastructure costs and improved efficiency
- Can drive down compute, network, and storage costs by leveraging on-demand delivery of applications in a nonpersistent desktop architecture.
- Can be implemented in any supported VMware vSphere datastore.
- Works with existing infrastructure for flexible delivery to users, groups, or devices.
- Enables IT to use the most appropriate storage, including fast storage with high-read IOPS, such as VMware vSAN™.
- Allows IT administrators to deliver and manage applications in virtual desktops using less storage capacity than they would with Horizon 7 alone.

**App Volumes Components and Architecture**

App Volumes integrates a simple agent-server–database architecture into an existing View virtual desktop deployment in Horizon 7. Centralized management servers are configured to connect to deployed virtual desktops that run an App Volumes Agent. An administrator can grant application access to shared storage volumes for users or virtual machines or both.

**App Volumes Components**
The major components of App Volumes are described in Table 1.

<table>
<thead>
<tr>
<th>ICON</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="App Volumes user" /></td>
<td><strong>App Volumes user</strong> – Active Directory user or organizational unit (OU) account. User account must be assigned local administrator rights if you want users to install their own applications in a writable volume.</td>
</tr>
<tr>
<td><img src="image" alt="App Volumes Manager" /></td>
<td><strong>App Volumes Manager</strong> – A Windows Server system used as the Web console for administration and configuration of App Volumes, and tracking of assignments for AppStacks and writable volumes.</td>
</tr>
<tr>
<td><img src="image" alt="App Volumes Database" /></td>
<td><strong>App Volumes Database</strong> – A Microsoft SQL (production) or SQL Express (nonproduction) database that contains configuration information for AppStacks, writable volumes, users, machines, assignments, and VM and user transactions.</td>
</tr>
<tr>
<td><img src="image" alt="App Volumes Agent" /></td>
<td><strong>App Volumes Agent</strong> – Software installed on all Windows desktops where users receive AppStack and writable volume assignments. The agent runs as a service and uses a filter driver to handle application calls and file-system redirects to AppStack and writable volume virtual machine disks (VMDK).</td>
</tr>
<tr>
<td><img src="image" alt="AppStack" /></td>
<td><strong>AppStack</strong> – A read-only volume containing any number of Windows applications, files, folders, registry settings, and more. Multiple AppStacks can be delivered to an individual system or user. An individual AppStack can also be delivered to more than one system or user.</td>
</tr>
<tr>
<td><img src="image" alt="Writable volume" /></td>
<td><strong>Writable volume</strong> – A user-specific read-write volume where the user is allowed to preserve application files and user-installed applications, settings, licensing information, and data. A user can have only one writable volume attached at a time to a desktop, but can have multiple writable volumes assigned.</td>
</tr>
<tr>
<td><img src="image" alt="Provisioning desktop" /></td>
<td><strong>Provisioning desktop</strong> – A clean desktop virtual machine that includes the OS and necessary updates and service packs, and has core applications installed. This machine acts as a master device that is used to install new applications on the AppStack. The provisioning virtual machine must have the App Volumes Agent installed and configured to connect to the App Volumes Manager.</td>
</tr>
<tr>
<td><img src="image" alt="vCenter Server" /></td>
<td><strong>vCenter Server</strong> – VMware vCenter Server® provides centralized management of vSphere virtual infrastructures. App Volumes leverages VMware vCenter Server for inventory information and operational connectivity to host, virtual machine, and storage resources within a deployed vSphere environment. For more information about using vCenter Server, see the <a href="link">vCenter Server and Host Management Guide</a>.</td>
</tr>
</tbody>
</table>

Table 1: App Volumes Components
Architecture of an App Volumes Deployment

Figure 2 shows the architectural layout of an App Volumes deployment with View virtual desktops in Horizon 7, with two data centers (North and South). This architecture can be extended to more data centers. The diagram also shows VMware User Environment Manager integrated into the deployment.
Figure 2: Multi-Data-Center Deployment of App Volumes with User Environment Manager

This diagram shows two data centers, each with the components necessary for a View desktop implementation using App Volumes. Each data center contains a View pod, which includes multiple View desktop blocks and a single View management block. The single clustered App Volumes SQL database can support all the App Volumes Managers in the environment and spans multiple View desktop blocks, as well as multiple View pods. The single App Volumes SQL database defines an instance of App Volumes.

A single App Volumes instance can communicate with all the vCenter Servers in both of the View desktop blocks, as shown in the diagram.

Each View desktop block consists of a VM cluster of desktop and RDSH pools, VMware ESXi™ hosts, shared storage for VMs, a switched Ethernet network, and a vCenter Server. Each View desktop block can have up to approximately 2,000 sessions.

Each View desktop block is supported by a View management block. The management block includes vCenter Server, Access Point appliances, View Connection Servers, GPOs for User Environment Manager, VMware vRealize® Operations for Horizon, and the App Volumes Managers.

CIFS shares store VMware ThinApp® packages, user data through folder redirection, and User Environment Manager profiles and configuration files.

Shared storage has separate datastores for View master images and for App Volumes AppStacks. The detail on the AppStack storage groups shows a storage group of shared datastores for each data center, with a non-attachable datastore for AppStack replication across data centers.

View master images, User Environment Manager profiles and configuration files, and user data on CIFS shares are replicated between the data centers.

VMware User Environment Manager combined with folder redirection is the recommended solution in this design for user persona management in a multi-data-center deployment. For more information, see the VMware blog post VMware User Environment Manager with VMware App Volumes.

This design shows how AppStacks are replicated between the two data centers using the non-attachable datastore and App Volumes storage groups as the mechanism to replicate the AppStacks. The design does not include writable volumes that can optionally be used to support user-installed applications. For more information on App Volumes multi-site design and how to replicate writable volumes, see the VMware Horizon 7 Enterprise Edition Multi-Site Reference Architecture.

Data Flow in an App Volumes Environment

Figure 3 shows the data flow for App Volumes in a typical View environment. The diagram illustrates a View pod in a single data center with multiple View blocks.
Figure 3: App Volumes Data Flow

The left side of Figure 3 illustrates App Volumes without the direct-to-host connection option (Mount on Host). The right side illustrates App Volumes with the direct-to-host connection option. The direct-to-host-connection option is recommended for deployment at scale because AppStacks and writable volumes are mounted more quickly when ESXi directly mounts them, rather than waiting for vCenter Server to direct ESXi to mount the volumes.
Note: The direct-to-host connection requires that all ESXi hosts use the same login and password for App Volumes operations. Root is not required and a custom role can be used to set up this connection.

For both connection options, the user logs in to their View desktop, and the connection is authenticated through the broker, the View Connection Server. The App Volumes Agent on the View desktop connects to the App Volumes Manager via a load balancer. The App Volumes Agent checks with the App Volumes Manager whether the user has a writable volume and AppStacks to mount. The App Volumes Manager retrieves this information from the App Volumes SQL database.

Without the direct-to-host-connection option, the App Volumes Manager requests that vCenter Server mount the user’s assigned volumes. VMware vCenter Server directs ESXi to mount the volumes. This attachment of volumes to the desktop begins during user login to the desktop. By the time the user logs in, their volumes are attached to their desktop. Not using the direct-to-host-connection option is adequate in small-scale deployments. For production deployments, we recommend the direct-to-host option in order to minimize the load on vCenter Server and eliminate potential performance bottlenecks.

With the direct-to-host-connection option, the App Volumes Manager directs ESXi to mount the volumes. This attachment of volumes to the desktop begins during user login to the desktop. While the user logs in, their volumes are attached to their desktop. The benefit of the direct-to-host-connection option is that vCenter Server is not part of the data flow. This is particularly useful in large-scale deployments, where vCenter Server might get overloaded and create a bottleneck.

The dashed blue line in the diagram indicates an ongoing, periodic communication between the App Volumes Manager and vCenter Server. The App Volumes Manager finds out if the host is up, if storage is still available, and if VMs are still online. App Volumes Manager then informs vCenter Server which volumes to attach to the desktop. This background communication is essential in all deployments, to ensure that the current status of the environment is communicated to the App Volumes Manager. App Volumes Manager periodically checks with vCenter Server to find out what has changed in the environment.

Managing Applications with AppStacks

App Volumes simplifies application management for virtual desktop infrastructure (VDI), Remote Desktop Session Host (RDSH) implementations, and VMware ThinApp application life-cycle management. The App Volumes real-time application model separates IT-managed applications and application suites into application containers called AppStacks.

AppStack Design

AppStacks are read-only volumes containing applications that can be assigned to AD user accounts, groups, OUs, or computer accounts to enable delivery of applications to end users.

Administrators can combine core applications into a single AppStack, making it easy to assign applications to users through AD object assignment. Administrators can make application updates available immediately or on next login or reboot.

Administrators can also manage line-of-business or departmental applications by combining them into other AppStacks, which are managed and deployed separately from the core AppStack.

Administrators can also choose to create single-application AppStacks for granular control or licensing restrictions. However, the preferred method is to combine multiple applications in a single AppStack.

Different AppStack designs can be mixed and matched to make application deployment and management easier and more efficient.

Application Suitability

Most applications work well with App Volumes, with little to no additional interaction needed. The following is a brief description of situations and application types where App Volumes may need special attention to work properly or where the application would work best within the master image, rather than in an AppStack. This section addresses these special cases in detail.

- **Just because you can, does not mean that you should** – Applications that should be available to the OS in the event that an AppStack or writable volume is not present should remain in the master image and not in an App Volumes container. 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 master image.

- **Tightly OS-integrated applications** – Applications that integrate tightly with the OS should not be virtualized in an AppStack. 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 master image and not in an AppStack. 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.

- **Know your app** – 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.

- **“Garbage in, garbage out”** – 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 provisioning process and then delivers that installation. If the installation is not accurate or configured correctly, then the delivery of that application will also be incorrect. It is important to verify and test the installation process to ensure a consistent and reliable App Volumes delivery.

- **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, then the App Volumes mini-filter driver would never get a chance to process the request, and thus would 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 disabled or removed from the OS while you install applications with App Volumes. There might be additional scenarios where App Volumes Agent should be disabled, allowing other applications to install correctly in the base OS.

- **User profile as part of an application install** – App Volumes does not capture the user profile space (`C:\users\<username>`). If an application as part of its installation process places components into this space, it will not be recorded as part of the provisioning process. If this happens, undesired consequences or failure of the application might result when the application is captured in an AppStack.

- **Additional application virtualization technologies** – Other application virtualization technologies (Microsoft App-V, VMware ThinApp, and others) should be disabled during provisioning because the filter drivers could potentially conflict and cause inconsistent results in the provisioning process.

- **Mixing of 32- and 64-bit OS types** – The OS type (32- or 64-bit) of the machine that the AppStack is attached to should match the OS type that applications were provisioned 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** – Entire applications may not work when installed on an App Volumes AppStack. There is no list of such applications, but an administrator may run into 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 later by identifying potential problems early, by looking at the application type and use case before deciding to create an AppStack.

**ThinApp Integration**

Network latency is often the limiting factor for scalability and performance when deploying ThinApp packages in streaming mode. Yet, ThinApp provides exceptional application-isolation capabilities. With App Volumes, administrators can present ThinApp packages as dynamically attached applications located on storage instead of moving bits around the data center over the network. This removes the network latency due to Windows OS and environmental conditions. Using App Volumes to deliver ThinApp packages allows for the best of both worlds—real-time delivery of isolated and troublesome applications alongside other applications delivered on AppStacks.

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 an AppStack using all the storage options available for use with AppStacks. This architecture permits thousands of virtual desktops to share a common ThinApp package via AppStacks without the need to stream or copy the package locally.
RDSH Integration
App Volumes supports AppStack integration with Microsoft RDSH shared desktops and published applications.

RDSH integration is as easy as deploying the App Volumes Agent on the RDSH host. After the agent is deployed, AppStack assignment is handled through the machine assignment process (by computer name) in the App Volumes Manager interface.

RDSH servers receive machine assignments of AppStacks rather than user assignments. AppStacks are attached to the RDSH host. Users are entitled to the RDSH desktops or applications through the View entitlement process. Writable volumes are not supported with RDSH assignments.

Microsoft Windows Server 2008 R2 and 2012 R2 are supported for RDSH use cases with App Volumes.


Microsoft Office Applications on AppStacks
For deploying Microsoft Office applications through App Volumes, see the VMware knowledge base article VMware App Volumes 2.x with Microsoft Office Products (2146035).

AppStack Templates
By default, a single 20-GB AppStack template is deployed in an App Volumes environment. This template is thin provisioned and provided in both a VMDK and VHD format. This template can be copied and customized depending on how large the AppStack needs to be for a given deployment scenario. For more information, see the VMware blog post Creating a Customized App Volumes AppStack Template VMDK.

If you have AppStacks from a previous 2.x release of App Volumes, they will continue to work with App Volumes 2.12.

App Volumes Writable Volumes
The App Volumes writable volumes feature enables the creation of a per-user volume where the following user-centric data can be installed and configured in different ways and move with the user:

- Application settings
- Licensing information
- Configuration files
- User-installed applications

Writable volumes do not provide a complete user environment management solution, but complement a user environment management solution. VMware User Environment Manager is a companion to App Volumes and provides management of user application settings that are applied when the user logs in or when an application launches. VMware User Environment Manager can manage data within writable volumes at a more granular level and provide contextual rules to enforce policies based on different conditions or events. For technical details on combining App Volumes and User Environment Manager, see the VMware blog post VMware User Environment Manager with VMware App Volumes.

Note the key differences between AppStacks and writable volumes:

- AppStack VMDKs are mounted as read-only and can be shared among all desktop virtual machines (VMs) within the data center.
- Writable volumes are dedicated to individual users and are mounted as the user authenticates to the desktop. Writable volumes are user-centric and roam with the user for nonpersistent desktops.

After writable volumes are created and assigned to a user, that user can install and configure applications. For this functionality to work properly, users require account permissions that allow application installation. App Volumes defers to Microsoft Windows security policies to determine user rights assignments.
Writable Volume Templates

By default, two writable volume templates are available when App Volumes Manager is installed:

- **UIA Only** – User-installed applications only
- **UIA and User Profile**

The UIA Only writable volume template is intended to capture user-installed applications. It is for use when you are combining App Volumes with a user-profile solution, such as VMware User Environment Manager, or a third-party user-profile solution. This template does not capture the user-profile data.

The UIA and User Profile writable volume template is intended to capture user-installed applications and the local profile data of the assigned user.

If you have writable volumes from a previous 2.x release of App Volumes, they will continue to work with App Volumes 2.12.

Expandable Writable Volumes

If a user’s writable volume has reached or is about to reach full capacity, it can be expanded.

Allowing End Users to See the Size of Writable Volumes

You can view space remaining in writable volumes from the App Volumes Manager. You can also allow the end user to see available space on their writable volume, from their system volume. To do this, 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.
3. Within the Parameters key, create a new key called ReportSystemFreeSpace with a DWORD value of 0 (zero).

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

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

Following is a screenshot of the registry, with the proper path, key, and value created.

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 the end user looks at space through Windows Explorer before you make the registry changes, they can see the free space and total space reported for the C: drive.
After you make the registry modifications, and you reboot the system, the `C:\` object now reports free space on the user's writable volume (which is 9.41 GB total in this case). (Notice, however, that the total space still reflects the `C:\` 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**

To expand the writable volume for each user from the App Volumes Manager, locate the user’s writable volume in the Volumes tab under the Writables sub-tab, and expand the information on the specific writable volume. Click the Expand Volume option and enter a larger value in 1-GB increments. 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 again.

**Writable Volume Policies**

You can apply three basic but very useful policies when you create writable volumes in the App Volumes Manager interface.

- **Block Login** – The Block Login option prevents users from logging in without their writable volume. The App Volumes Manager prevents users from logging in to any additional computers when their writable volume is attached elsewhere. Administrators can toggle the Block Login option on each writable volume.

- **Limit Attachment** – The Limit Attachment option enables administrators to specify the prefix of a computer name that the writable volume can be attached to. When a prefix is specified, the writable volume can be attached only to a computer with a name that begins with that prefix.

- **Defer Create** – The Defer Create option defers the creation of writable volumes for group and OU members until their next login. This option affects only AD groups and OUs. Volumes for users and computer entities are created when a group or organizational unit is selected, a writable volume is created for each current member. These containers can have hundreds or thousands of members. However, creating a large number of volumes at once can take a long time, and in many cases, not every member needs a writable volume.

**Storage Groups**

Storage groups can be used to group datastores together. The two types of storage groups are:
AppStack storage groups – Used for replication.

Writable volume storage groups – Used for distribution.

**AppStack Storage Groups**

AppStack storage groups are used to replicate single AppStacks across multiple datastores. Using AppStack storage groups allows you to better manage storage-transaction aggregation and concurrent user-file maximums.

Two automation options for AppStack storage groups are available:

- **Automatic replication** – Any AppStack placed on any datastore in the storage group will be replicated across all datastores in the group every 4 hours.
- **Automatic import** – After replication, the AppStack will be imported into App Volumes Manager and be available for assignment from all datastores in the storage group.

When using AppStack storage groups, the App Volumes Manager manages the connection to the relevant AppStack, based on location and number of attachments across all the datastores in the group.

**Non-Attachable AppStack Datastores**

In App Volumes 2.10, the concept of a non-attachable datastore for AppStacks was introduced. When this option is selected for a specific datastore, any AppStacks placed on that datastore will not be attachable. This means that you can use a datastore on slow or inexpensive storage for replication, and use high-speed, low-latency storage for storing mountable volumes. This non-attachable datastore can also be used as a staging area for AppStack creation before deploying to production storage groups.

**Writable Volume Storage Groups**

Writable volume storage groups are used to distribute volumes across datastores for I/O and file distribution.

There are two writable-volume distribution strategy options:

- **Spread** – The goal is to distribute files evenly across all storage locations. When a file is created, it is stored on the datastore with the most available space.
- **Round-robin** – The goal is to distribute files sequentially across the storage locations. When a file is created, it is stored on the datastore with the oldest used time.

Note: Distribution strategies are not used for AppStacks. These options apply only to writable volumes.

**Deployment Requirements**

The specific requirements necessary to support an App Volumes deployment are listed in Table 2. See the App Volumes Release Notes for specific version and patch compatibility.
## App Volumes Deployment Considerations

### Table 2: App Volumes Deployment Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| **AD and vCenter Server user accounts**          | - A read-only AD account for each domain that App Volumes Agents will reside in.  
- To integrate App Volumes with vCenter Server, you must create a service account within a vCenter Server with administrator privileges. Optionally, you can create a service account with privileges granted by a custom user role.  
- A designated AD user group that allows administrator access to the App Volumes Manager server.  
- End users require administrator privileges to install applications into their writable volumes, if used. |
| **App Volumes Manager server**                   | - Windows Server 2008 R2 or Windows Server 2012 R2  
- .NET 3.5 Framework deployed  
- 2 vCPUs (4 vCPUs recommended)  
- 4 GB of RAM  
- 1 GB of disk space |
| **Hypervisor**                                   | - VMware ESXi 5.5 or 6.0 with vCenter Server 5.5 or 6.x  
- For VMware vSphere vMotion®, supported since App Volumes 2.10: ESXi 5.5 U3 or ESXi 6.0 U2 |
| **App Volumes Agent**                            | - Windows 7, Windows 8.1, or Windows 10, version 1607  
- For VDI or RDSH: Windows Server 2008 R2 or 2012 R2  
- Application Server role must be enabled for RDSH support  
- 1 GB of RAM  
- 5 MB of disk space |
| **App Volumes Manager database**                 | - SQL Express 2008 R2 supported for testing or nonproduction  
- SQL 2008 R2 or 2012 Standard  
- SQL Server account with DB Owner role required for installation and operations |
| **Supported browsers to access management console**| - Internet Explorer  
- Firefox  
- Safari  
- Chrome |
| **Active Directory**                             | Microsoft AD domain, 2003 functional or later |
| **Licensing**                                    | Valid VMware App Volumes license deployed on App Volumes Manager server(s) at time of App Volumes software installation |
| **Basic networking access**                      | The following default settings can be customized:  
- App Volumes Web console: 80 (HTTP), 443 (HTTPS)  
- App Volumes Agent-to-server communications: 80 (HTTP), 443 (HTTPS)  
- App Volumes server to vCenter Server and ESXi hosts: 443 (SOAP/hosted)  
- App Volumes server to remote SQL Server: 1433 |
| **Storage**                                      | - Shared storage for all ESXi hosts that host desktops connecting to AppStacks and writable volumes. Local host storage can be used for small test environments.  
- If local storage is used, AppStacks can be placed on the same datastore where VMs reside. The App Volumes service can be configured to refer to local storage before referring to shared storage. If specified AppStacks exist on the local storage, then these AppStacks are mounted in favor of duplicate copies that exist on shared storage. ** |
Planning for Production

Planning a production App Volumes deployment requires considerable care. Consider at least the following factors:

- Storage planning
- Performance
- Scalability
- Security
- Availability
- Recoverability

Storage Planning

Proper planning for storage can affect many aspects of an App Volumes implementation, including cost and performance. The most important considerations are described in the following sections.

AppStack and Writable Volume Template Placement

When new AppStacks and writable volumes are deployed, the templates are used as the copy source. Administrators should place these templates on a centralized shared storage platform. As with all production shared storage objects, the template storage should be highly available, resilient, and recoverable.

Free-Space Considerations

AppStack sizing and writable volume sizing are critical for success in a production environment. AppStack volumes should be large enough to allow applications to be installed and should also allow for application updates. AppStacks should always have at least 20 percent free space available so administrators can easily update applications without having to resize the AppStack volumes.

Writable volumes should also be sufficiently sized to accommodate all users’ data. Storage platforms that allow for volume resizing are helpful if the total number of writable volume users is not known at the time of initial App Volumes deployment.

Because all AppStacks and writable volumes are vSphere VMFS thin-provisioned, storage space is not immediately consumed. Follow VMware best practices when managing thin-provisioned storage environments. Free-space monitoring is essential in large production environments.

Writable Volumes Delay Creation Option

Two policy options can complicate free-space management for writable volumes:

- The option to create writable volumes on the user’s next login means that storage processes and capacity allocation are impacted by user login behavior.
- The option to restrict writable volume access (and thus initial creation) to a certain desktop or group of desktops can also mean that user login behavior dictates when a writable volume template is copied.

In a large App Volumes environment, it is not usually a good practice to allow user behavior to dictate storage operations and capacity allocation. For this reason, it is recommended to create writable volumes at the time of entitlement, rather than deferring creation.

Mount on Host

The Mount on Host option in the App Volumes Manager hypervisor settings allows App Volumes to issue an AppStack or writable volume mount command directly to the vSphere hosts instead of issuing the command to vCenter Server. Mount on Host offloads the demands on vCenter Server and provides resiliency if vCenter Server is temporarily offline or unavailable.

See the App Volumes Manager Storage Policy Best Practices for more information.

Note: For the Mount on Host feature to work, all vSphere hosts must have the same user credentials. Root-level access is not required. See the App Volumes documentation for the correct procedures on creation of a custom role with the required permissions.
Mount Local

If local storage is used, AppStacks can be placed on local datastores of the hosts where the VMs reside. The App Volumes policy can be configured to point at local storage before referring to shared storage. If the specified AppStacks are present on local storage, then these locally stored AppStacks are mounted instead of duplicate copies that exist on shared storage.

Note: For the Mount Local feature to work, all vSphere hosts must have the same user credentials. Root-level access is not required. See the App Volumes documentation for the correct procedures on creation of a custom role with the required permissions.

Performance

Test AppStacks immediately after provisioning to determine their overall performance. Using a performance analytics tool, such as VMware vRealize Operations Manager™, gather virtual machine, host, network, and storage performance information for use when AppStacks 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. AppStacks 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 AppStacks.

Writable volumes are read-write. Storage utilization patterns are largely influenced by user behavior with regard to desktop logins and logouts, user-installed applications, and changes to local user profiles. Group each set of similar users into use cases, and evaluate performance based on peak average use.

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

Host configurations have significant impact on performance at scale. Consider all ESXi best practices during each phase of scale-out. To support optimal performance of AppStacks and writable volumes, give special consideration to the following host storage elements:

- Host storage policies
- Storage network configuration
- HBA or network adapter (NFS) configuration
- Multipath configuration
- Queue-depth configuration

For best results, follow the recommendations of the relevant storage partner when configuring hosts and clusters.

Scalability

App Volumes performs very well at scale if you pay attention to recommendations. To begin, see the VMware App Volumes Reference Architecture for results on capacity and performance testing of App Volumes in a View environment.

View and vSphere at Scale

In a production View environment, it is important to adhere to the following best practices:

- Sizing and scaling best practices, as documented in the View Architecture Planning guide
- vSphere scalability best practices, as documented in the Performance Best Practices for VMware vSphere 6.0
- vSphere architectural best practices, as documented in the vSphere Resource Management Guide

Multi-vCenter-Server Considerations

Configuring multiple vCenter Servers is a way to achieve scale for a large View pod, for multiple data centers, or for multiple sites.

In a multiple-vCenter-Server environment, an AppStack is tied to storage that is available to each vCenter Server. It is possible that an AppStack visible in App Volumes Manager could be assigned to a VM that does not have access to the storage.

To avoid this issue, use storage groups to replicate AppStacks across vCenter Servers. For more information about storage groups in general, see the VMware App Volumes User Guide.

The VMware Horizon 7 Enterprise Edition Multi-Site Reference Architecture discusses how to use storage groups to replicate AppStacks across sites and between vCenter Server instances.
Multiple Active Directory Considerations

App Volumes supports environments with multiple Active Directory domains with and without the need for trust types configured between them. An administrator can add multiple Active Directory domains through the Configuration > Active Directories tab. An account with a minimum of read-only permissions for each domain is required. You must add each domain that will be accessed for App Volumes by any computer, group, or user object. In addition, non-domain-joined entities are now allowed by default. See the App Volumes User Guide for more information.

App Volumes SQL Database at Scale

An App Volumes instance is defined by the SQL database. Multiple App Volumes Manager servers are connected to a single SQL database. The database contains all the configuration information for the entire App Volumes instance.

For nonproduction App Volumes environments, you can use the Microsoft SQL Server Express database option, which is included in the App Volumes Manager installer. Production App Volumes environments should use an Enterprise or Standard edition of Microsoft SQL. When designing the SQL environment that supports App Volumes, it is important to follow Microsoft best practices. SQL limits, not App Volumes limits, apply to the number of objects per database.

Do not use SQL Server Express at large scale. Place SQL Server on its own dedicated VM and ensure there is adequate performance capability to support the SQL instance. For more information on SQL Server performance in an App Volumes instance, refer to the VMware App Volumes Reference Architecture.

App Volumes works well with both Always On Failover SQL clusters and Always On Availability Groups. Your SQL DBA or architect will decide which option better fits your environment.

App Volumes Manager Servers at Scale

A single App Volumes Manager server has been tested with up to 2,000 managed users. You can address the additional performance load that comes from an environment that has thousands of users by adding a load-balanced cluster of App Volumes Managers.

Use at least two App Volumes Managers in production and configure each App Volumes Agent to point to a load balancer, or use a DNS server that resolves to each App Volumes Manager in a round-robin fashion.

As a best practice, we recommend not building App Volumes Managers to handle the maximum tested user load of 2,000. It is recommended to build to scale so that each App Volumes Manager server is handling 1,000 to 1,500 users.

App Volumes Manager servers have been tested to service one user login per second. This means that in a high-load login event where more than one user per second is logging into a single App Volumes Manager, login times may increase. This is similar to the login rate of View Connection Servers.

AppStacks at Scale

The current best practice is to limit each virtual desktop to no more than 15 attached AppStacks. Although App Volumes can support more attached AppStacks up to the vSphere limit for VMDKs attached to a VM, there can be performance implications when attaching more AppStacks. In production environments, it is a good practice to combine multiple applications into each AppStack.

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

- Do not place AppStacks and VMs on the same datastore.
- For best performance, limit concurrent attachments of a single AppStack VMDK to 750 – 1,000 VMs.
- Use storage groups for AppStacks when they 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 vSAN:

- AppStacks and VMs can be placed on a single datastore.
- Concurrent attachments of a single AppStack can be increased beyond the previous guidance of 750 – 1,000 VMs. It is best to benchmark performance to ensure a good user experience.
- Storage groups for AppStacks are not applicable in a vSAN implementation.

In large-scale deployments, it is useful to enable the direct-to-host-connection option (Mount on Host).
Security
To support a large production environment, there are some important security configurations that administrators should put into place:

- Open only essential, required firewall ports on App Volumes Manager and SQL. Consider using an advanced firewall solution, such as VMware NSX®, to dynamically assign virtual machine firewall policies based on server role.
- Replace the default self-signed SSL certificate with a certificate for App Volumes Manager signed by a reliable certificate authority.
- The App Volumes Manager communicates with vCenter Server over SSL. The manager must trust the vCenter Server certificate. The manager can be configured to accept an unverifiable (self-signed) certificate through Configuration > Machine Managers. Each machine manager (vCenter Server) has a Certificate option that will show the current status of the certificate and will allow an administrator to explicitly accept an unverifiable certificate.
- Consider using ThinApp to package applications, to take advantage of the security benefits of isolation modes, when required. Each ThinApp package can be isolated from the host system, and any changes, deletions, or additions made by the application to the file system or registry are recorded in the ThinApp sandbox instead of in the desktop operating system. For more information, see Thin App Resources.
- Writable volumes with user-installed applications require that each desktop user be assigned local computer administrator privileges to allow the installation and configuration of applications. Some use cases could benefit from temporary, request-based elevated privileges to allow incidental application installation for a specific user or user group. Carefully consider the security risks associated with granting users these elevated privileges.
- It is good security hygiene and a forensics best practice to create and use an AD user service account specifically for App Volumes. It is never a good idea to use a blanket or general purpose AD user account for multiple purposes within AD.
- Administrators might choose to create an administrative role in vCenter Server to apply to the App Volumes service account.

Availability
As with all server workloads, it is strongly recommended that enterprises host App Volumes Manager servers as vSphere virtual machines. VMware vSphere availability features such as cluster HA, VMware vSphere Replication™, and VMware Site Recovery Manager™ can all complement App Volumes deployments and should be considered for a production deployment.

In production environments, avoid deploying only a single App Volumes Manager server. It is far better to deploy an enterprise-grade load balancer to manage multiple App Volumes Manager servers connected to a central, resilient SQL database instance.

As with all production workloads that run on vSphere, underlying host, cluster, network, and storage configurations should adhere to VMware best practices with regard to availability. See the vSphere Availability Guide for more information.

Recoverability
As with any enterprise solution, App Volumes must be included in business continuity and disaster-recovery planning.

The VMware Horizon 7 Enterprise Edition Multi-Site Reference Architecture provides information on planning for multi-site operation and recovery of App Volumes.

Deployment Procedures
The following diagram shows the basic steps necessary for a typical App Volumes deployment.
Figure 4: App Volumes Deployment Steps

To deploy App Volumes:

1. Install the App Volumes Manager.
2. Configure the App Volumes Manager.
3. Install the App Volumes Agent on desktops.
4. Build a provisioning virtual machine.
5. Create AppStacks and writable volumes.

Install App Volumes Manager

For the App Volumes Manager, choose a new Windows Server 2008 R2 or Windows Server 2012 R2 and ensure it is a domain member and is configured with a static IP address. Use the Windows Server Manager to verify that the server is configured with the Application Server role and that the .NET 3.5 Framework is deployed.

Create a new database instance for App Volumes. For small environments, a local SQL Express instance can be deployed upon installation. SQL Express is recommended only for small, nonproduction deployments. Configure and use a SQL account with DB Owner privileges for installation and operation.

The installer for App Volumes Manager is easy to use and allows customization of the network ports and the installation directory during the installation process. The application installation consumes approximately 200 MB of server space.

Starting with App Volumes 2.12, port 80 is disabled by default during installation. If you would like to enable port 80, you will need to select Allow connections over HTTP (insecure).

Configure App Volumes Manager

After the App Volumes Manager installation is complete, restart the server. On the first App Volumes Manager launch after restart, provide configuration details, such as:

- License key
- AD information
- Whether to allow non-domain users and desktops to connect to AppStacks and writable volumes
- AD user group for administrator access to App Volumes Manager
- vCenter Server instance and administrator-level credentials
- Default storage location for both AppStacks and writable volumes

For every App Volumes Manager installation that needs to support vSphere vMotion (only supported on ESXi 5.5 Update 3 and ESXi 6.1 or later), the AVM_PROTECT_VOLUMES environment variable must be set to 1. Follow these steps on each App Volumes Manager server:

1. Open the Control Panel and click System Properties.
2. Click the Advanced tab, and click Environment Variables.
3. Under the System Variable section, click New.
4. For Variable, enter AVM_PROTECT_VOLUMES.
5. For **Value**, enter 1.

When the server configuration is complete, deploy the App Volumes Agents. For additional details, see the [VMware App Volumes User Guide](#).

**Install App Volumes Agent**

Deploy App Volumes Agents on desktops that access AppStacks and writable volumes, as well as on provisioning machines (desktops used to capture applications for AppStacks) and RDSH hosts.

![Diagram of App Volumes Agent Relationships](image)

**Figure 6: App Volumes Agent Relationships**

The App Volumes Agent installation process is the same for client desktops, RDSH servers, and provisioning VMs. The same installer used for App Volumes Manager is used for agents. Select the **Agent** option after starting the installer. During the installation process, the following information is required:

- Host name or IP address for the App Volumes Manager that the agent needs to connect to. If multiple App Volumes Managers are deployed, enter the FQDN or virtual IP address for a load balancer that services a group of App Volumes Manager servers.
- TCP port (default 80) for client-server communications.
The installer prompts for a required OS reboot after agent installation is complete. After the agent is installed on a provisioning virtual machine, you must take a snapshot of the provisioning machine to be able to later revert to a clean state.

Provisioning machine creation is described in **Build a Provisioning Virtual Machine**. See **Best Practices for Client Desktops** for more information about the App Volumes Agent.

**Build a Provisioning Virtual Machine**

To provision applications to an AppStack, you need a clean provisioning machine for installation of applications and subsequent transfer of those applications to the AppStack. To build a provisioning virtual machine:

1. Prepare the provisioning machine.
2. Install the App Volumes Agent on the provisioning machine, and reboot.
3. Take a snapshot of the provisioning machine.

**Prepare the Provisioning Machine**

The virtual machine used to provision AppStacks should be almost identical to the virtual machines deployed to users, ideally from the same master image and domain-joined.

Although not strictly required, it is a recommended best practice to use the same specific Windows OS for the provisioning machine that an AppStack will be attached to. For example, use a Windows 10 provisioning machine for AppStacks that will be attached to Windows 10 desktops, and a Windows Server 2012 R2 provisioning machine for AppStacks that will be attached to a Windows Server 2012 R2 RDSH host.

If you have any applications in your master image that you want to put into an AppStack, you must recreate the master image without the applications. The provisioning machine should not have any uninstalled applications that will be placed in an AppStack.

Any agents that will perform any type of update should be disabled, such as Windows Update and virus definitions.

Disable as many agents as possible, for example, antivirus agents, Horizon Agents, and additional application virtualization clients such as App-V.
Install the App Volumes Agent on the Provisioning Machine
To install the App Volumes Agent on the provisioning machine, refer to Install App Volumes Agent for instructions.

Take a Virtual Machine Snapshot
Take a snapshot of the provisioning virtual machine. This snapshot of the provisioning machine is in a clean state, before any applications for the AppStack are captured, and it allows you to return the provisioning machine to a ready-to-capture state.

Create AppStack and Writable Volume
Now that you have prepared the provisioning machine, you are ready to provision an AppStack and a writable volume.

AppStack Creation and Assignment
The following diagram gives an overview of the steps to create, provision, and assign an AppStack.

Create and attach the AppStack:
1. In the App Volumes Manager, on the AppStacks tab, click Create AppStack.
   a. Enter a name for the AppStack.
   b. Select a template from the drop-down menu.
   c. Enter the path of the AppStack.
   2. Click Create and confirm the AppStack creation.
3. If you click Rescan, you can see the newly created AppStack, and the status changes to Creating.
4. If you click Rescan again a minute later, the status changes to Unprovisioned.

Install applications in the provisioning desktop:
1. Select the AppStack you created, and click Provision.
2. In the Find Provisioning Computer field, enter a name or click Search to find your provisioning computer.
3. Select the one with status of Available, and click Provision.
4. Click Start Provisioning.
   Note: Local administrator rights may be required on the provisioning machine, depending on the applications being installed.
5. Log in to the provisioning machine.
   A dialog box appears that must stay open during all of the installations and reboots.
Note: If reboots are needed, the dialog box re-appears afterward. Do not click OK until all needed reboots have completed.
6. Install any application, and launch it to verify that it works.
7. Click OK only after installing all applications.

Complete AppStack creation:

1. After installing all applications, in the dialog box, click Yes to finish.
2. Click OK to reboot the provisioning desktop.
3. Log in again, and in the Provisioning successful dialog box, click OK.
   After successful provisioning, exit code 0 is displayed in the App Volumes Agent pop-up window.

Validate the AppStack object:

1. Verify that the App Volumes Manager lists the new AppStack as Enabled.
2. Revert the provisioning machine to a previously saved snapshot.

At this point, the AppStack is ready to be assigned to users.

Assign the AppStack to AD users:

1. Select the AppStack and click Assign.
2. In the Search Active Directory field, enter the user name or group name, and click Search.
3. Select the user name or group name, and click Assign.

For detailed steps to create, provision, and assign AppStacks, see Application Management with App Volumes in the VMware App Volumes Reviewer’s Guide.


Writable Volume Creation and Assignment

Writable volumes are read-write volumes that support user-installed applications. To create a writable volume, you copy a writable volume template and assign that writable volume to an Active Directory user, user group, or computer. After creation, writable volumes follow users from one computer to another, but cannot be reassigned to another user. Each user can have multiple writable volumes assigned but only one writable volume in use per session. For detailed instructions on creating writable volumes, see the VMware App Volumes User Guide.

Best Practices for an App Volumes Deployment in Production

This section contains a nonexhaustive list of best practices for deploying, maintaining, and operating a production VMware App Volumes environment.

Database Best Practices

When configuring a database for an App Volumes deployment in production:
Use at least a Microsoft SQL Standard instance, rather than the bundled SQL Express instance, which can be installed on the App Volumes Manager server. An Enterprise or Standard SQL Server database instance provides better performance, allows greater scalability, and enables easier backup and recovery. For more information, see the App Volumes Reference Architecture guide.

Make sure the Microsoft SQL Server is resilient and highly available. The App Volumes database represents a possible single point of failure. If the database is down or unreachable, App Volumes cannot deliver volumes.

If deploying multiple App Volumes Manager servers with a shared SQL database, after the initial App Volumes Manager deployment, deselect the Create a new database or overwrite existing database option during subsequent App Volumes Manager installations.

Master Image Best Practices

Master images should be optimized for VDI or RDSH to ensure the best performance possible in a virtualized environment. The VMware OS Optimization Tool helps optimize Windows 7, 8, 10, 2008, and 2012 systems for use with VMware Horizon 7. The optimization tool includes customizable templates to enable or disable Windows system services and features, per VMware recommendations and best practices, across multiple systems. Because most Windows system services are enabled by default, the optimization tool can be used to easily disable unnecessary services and features to improve performance.

Best Practices for Creating and Provisioning AppStacks

Consider the following best practices when creating and provisioning AppStacks:

- The following characters cannot be used when naming AppStacks:
  - & * ' < >
- Provision AppStacks on a clean master image that resembles as closely as possible the target environment where the AppStack is to be deployed. For example, the provisioning virtual machine and target should be at the same patch and service pack level and, if applications are included in the master image, they should also be in the provisioning virtual machine.

Caution: Do not use a provisioning machine where you have previously installed and then uninstalled any of the applications that you will capture. Uninstall may not clean up all remnants of previous applications, and the App Volumes application capture will not be complete.

- Perform provisioning on a virtual machine that has no AppStacks previously assigned to it. If any AppStacks have been assigned to the virtual machine, or the virtual machine has been used previously for provisioning, revert that virtual machine to the clean snapshot before provisioning a new AppStack.

Best Practices for Configuring AppStacks

When there is an application conflict, the last AppStack virtualized “wins.” In the App Volumes Manager, use the Override Precedence option. In the Directory tab, click the Users, Computers, or Groups sub-tab and select one of the objects. The Override Precedence option allows you to define AppStack ordering. It may be necessary to reorder AppStacks in order to remove application conflicts. This option can also be used to ensure that an AppStack with a supporting application loads before an AppStack with an application that requires that supporting application.

Best Practices for Updating AppStacks and Assigning Updated AppStacks

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

- After updating an AppStack, unassign the original AppStack before assigning the updated AppStack. Failure to unassign the old AppStack before assigning the new one can result in application conflicts because most Windows applications cannot run side by side with older versions of themselves in a Windows OS.
- Unassign AppStacks to take effect on next login rather than immediately. Removing applications while in use could result in user data loss and OS instability.

Sizing Best Practices for AppStacks

When sizing AppStacks:

- Limit each virtual desktop to no more than 15 attached AppStacks.
- The size of the default AppStack 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 VMware
Knowledge Base article Creating a New App Volumes AppStack template VMDK smaller than 20 GB (2116022).

- When deploying user-assigned AppStacks, use fewer AppStacks, rather than more. You can include more than one application per AppStack.

App Volumes Manager Storage Policy Best Practices

When configuring the App Volumes Manager storage policy, select the Mount on Host option, if possible. This option speeds up AppStack and writable volume mount operations and provides resiliency in a situation where vCenter Server is not online.

Note: You may not be able to use the Mount on Host option. This feature requires that the password for the ESXi account used by that App Volumes Manager be consistent across all ESXi hosts.

Best Practices for Client Desktops

When setting up client desktops, consider the following best practices:

- When reverting a desktop virtual machine that is running the App Volumes Agent to a previous snapshot, make sure that the virtual machine is gracefully shut down, to avoid syncing issues. This is primarily relevant to the provisioning desktop and master VMs for View linked- and instant-clone pools.
- If using a View pool, the App Volumes Agent should be installed on the master VM for linked- and instant-clone pools, or on the virtual machine template for full-clone pools, for ease of distribution.
- If using a View 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.

Miscellaneous Best Practices

Enable vCenter Server datastore browser options. This feature is enabled in the default configuration for vCenter Server. For more information, see the vSphere Hardening Guide.

Additional Resources

- App Volumes product documentation
- VMware vSphere 6.x documentation. See especially the vSphere Availability Guide.
- VMware vSphere 5.x documentation. See especially the vSphere Availability Guide.
- VMware vSphere security hardening guides
- VMware knowledge base article: VMware App Volumes 2.x with Microsoft Office Products (2146035)
- Implementation Considerations for VMware App Volumes in a Citrix XenApp Environment
- Implementation Considerations for VMware App Volumes in a Citrix XenDesktop Environment
- VMware App Volumes Reference Architecture
- VMware User Environment Manager product page
- VMware User Environment Manager with VMware App Volumes
- VMware Horizon 7 Enterprise Edition Multi-Site Reference Architecture
- Antivirus Considerations in a VMware Horizon 7 Environment
- VMware knowledge base article: VMware App Volumes order of operation (2143163)
- VMware App Volumes and Microsoft Office, Part 1: Choosing the Correct Office Version
- Replacing the Self-Signed Certificate in VMware App Volumes 2.12

About the Authors and Contributors

The following team wrote this document:

- Jason Marshall, Senior Manager, Product Engineering, VMware, updated the content and diagrams to App Volumes 2.12.
- Tina de Benedictis, Senior Manager, End-User-Computing Technical Marketing, VMware, worked with Jason Marshall to rewrite the paper for App Volumes 2.12.
- Tristan Todd, Senior Solutions Architect, VMware, wrote the original version of this paper and provided ongoing review and input as the paper was revised for the 2.12 release of the product.
- Stéphane Asselin, Senior End-User-Computing Architect, Customer Success Team, End-User Computing, VMware, provided input and ongoing review for the original paper and for the App Volumes 2.12 version.
Dean Flaming, Senior Technical Marketing Manager, Customer Success Team, End-User Computing, VMware, provided input and ongoing review for the original paper and for the App Volumes 2.12 version.

Jim Yanik, Senior Manager, End-User-Computing Technical Marketing, VMware, provided input and ongoing review for the original paper and for the App Volumes 2.12 version.

Samir Patel, VMware alumnus, provided input and ongoing review for the App Volumes 2.12 paper.

**Contributors**

- Gina Daly, Technical Marketing Manager, End-User-Computing Technical Marketing, VMware, contributed to the content of the App Volumes 2.12 paper.

To comment on this paper, contact VMware End-User-Computing Technical Marketing at euc_tech_content_feedback@vmware.com.