



# VMware App Volumes Deployment Guide

VMware App Volumes 2.5

TECHNICAL WHITE PAPER

## Table of Contents

Executive Summary . . . . .	3
App Volumes Overview. . . . .	3
How App Volumes Works. . . . .	4
Summary of App Volumes Benefits . . . . .	5
Architectural Overview . . . . .	6
Application Management with AppStacks . . . . .	8
Application Stacks . . . . .	8
Application Suitability. . . . .	8
ThinApp Integration. . . . .	9
RDSH Integration . . . . .	9
AppStack Templates . . . . .	9
Writable Volumes. . . . .	10
Writable Volume Templates . . . . .	10
Writable Volume Policies . . . . .	10
User Installed Applications. . . . .	11
Local User Profiles . . . . .	11
Writable Volume Storage Groups . . . . .	11
Deployment Overview. . . . .	12
App Volumes Agent . . . . .	13
Agent Installation . . . . .	13
Provisioning Desktop Creation . . . . .	14
Deployment Requirements. . . . .	16
Planning for Production . . . . .	18
Storage Planning . . . . .	18
Performance . . . . .	19
Scalability. . . . .	19
Security . . . . .	20
Availability . . . . .	20
Best Practices . . . . .	21
About the Author and Contributors . . . . .	22
References. . . . .	22

## Executive Summary

This deployment guide presents a high-level overview of [VMware App Volumes™](#). It describes App Volumes capabilities, architecture, and implementation requirements and addresses frequently asked high-level questions about deploying an App Volumes solution.

## App Volumes Overview

App Volumes is a real-time application delivery and lifecycle management tool. Enterprises can use App Volumes to build real-time application delivery systems that ensure that applications are centrally managed. Applications are delivered to desktops through virtual disks. There is no need to modify desktops or applications themselves, and the App Volumes solution can be scaled out easily and cost-effectively, without compromising end-user experience.

App Volumes complements the VMware End-User Computing portfolio by integrating with existing VMware Workspace™ Portal, application, and desktop solutions.

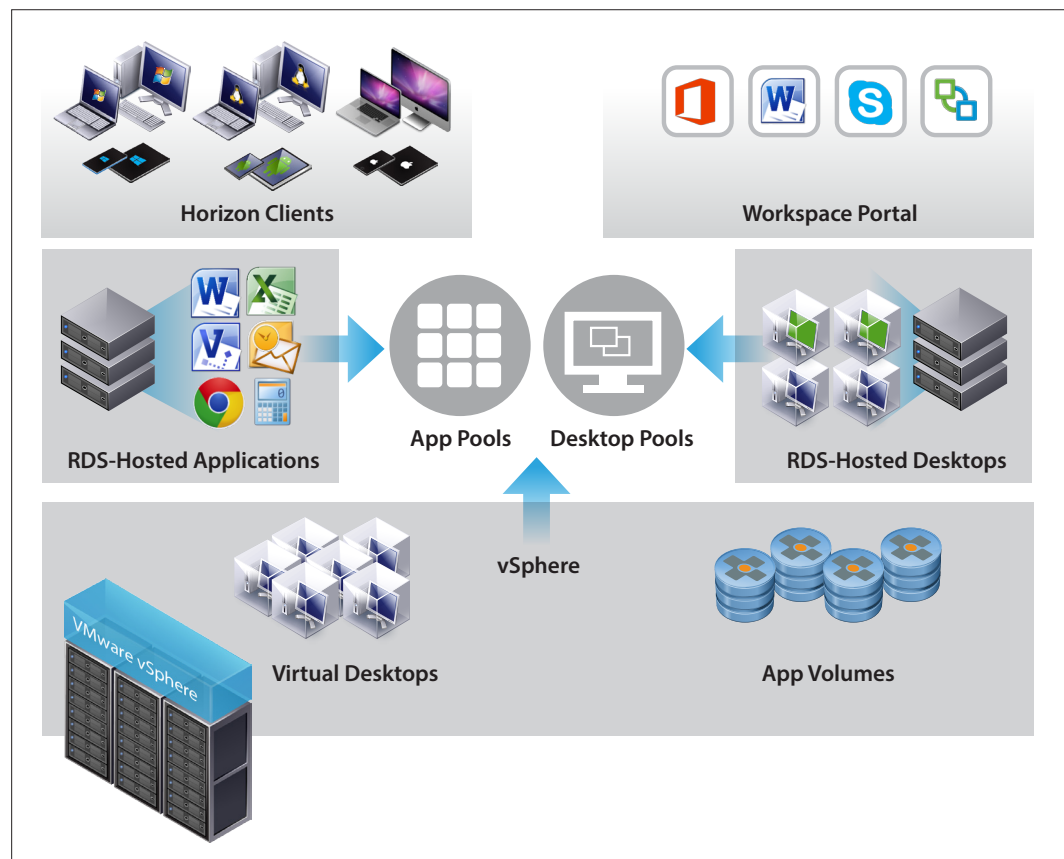
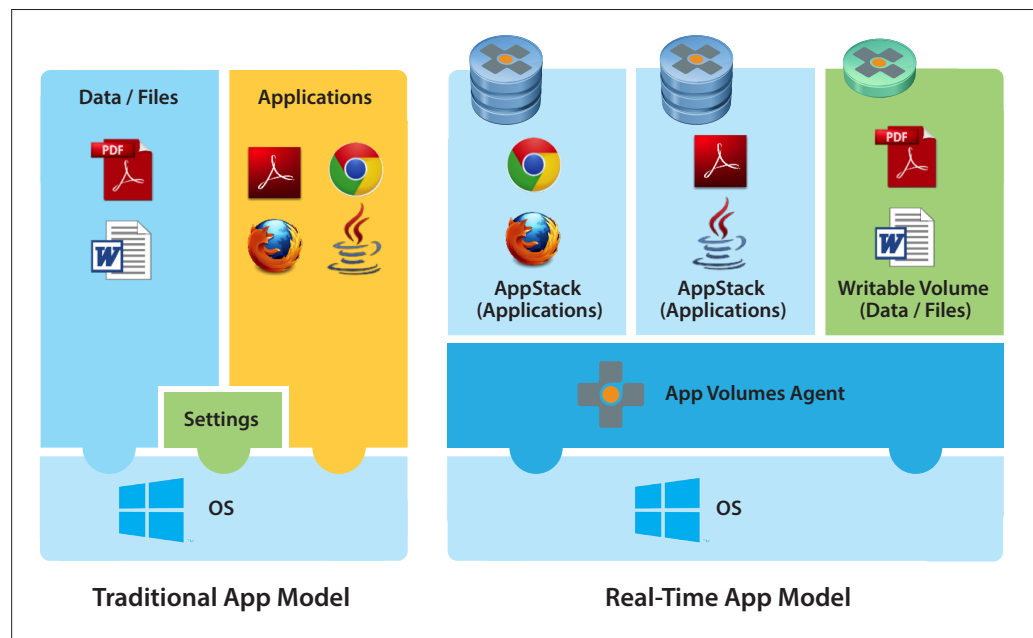


Figure 1: VMware End-User Computing Solutions

## How App Volumes Works

In modern desktop environments, the demand for real-time application delivery often puts strain on existing infrastructures. Through VMware App Volumes, VMware addresses this strain by offering an alternative for per-virtual machine and per-application management and by virtualizing applications above the OS. Applications virtualized above the OS are provisioned natively—without modification, in various configurations, to multiple groups of users—and, through file and registry virtualization, are organized into application management containers. This arrangement leverages existing storage and networking services, reduces infrastructure strain and overhead, and simplifies application lifecycle management.

As illustrated in Figure 2, a desktop operating system (OS) with an App Volumes agent places application management containers above the OS. Applications, data files, settings, middleware, and configurations act as separate, layered containers. These containers are called application stacks (AppStacks) when in read-only mode or writable volumes when in read-write mode.



**Figure 2:** Application Management Containers

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. AppStacks are stored in shared volumes across virtual disks. They require no special packaging formats or snapshot technologies and are easy to provision. During provisioning, App Volumes intelligently records the entire application installation process, and it makes the changes made by the native application installers available for delivery to users.

Administrators can easily deliver provisioned AppStacks to the system, or to a user or group. Applications delivered by App Volumes look and feel natively installed, and they follow users seamlessly 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.

Writable volumes enable a per-user, per-session volume where data such as Windows user profiles (including settings within local user profiles), files, and user-installed apps can be installed, configured, and set to persist across sessions. Per-user volume data moves across systems with the user. This complements user environment management (UEM) solutions, which can manage data within writable volumes at a more granular level, and provide contextual rules to enforce policy, based on different conditions or events.

## Summary of App Volumes Benefits

With App Volumes, applications become OS-agnostic 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 application delivery**

- Deliver and upgrade applications at scale and in seconds
- Dynamically deliver applications without interrupting users even if they are logged in
- Complete application lifecycle management (provisioning, delivery, maintenance, retirement)

- **Cost-optimized infrastructure**

- Optimized application delivery drives down compute, network, and storage costs
- Can reduce storage costs for VDI significantly
- Works with existing infrastructure with flexible delivery to users, groups, or devices

- **Seamless end-user experience**

- Support fully customizable desktops, with the freedom for end users to install their own applications
- Persistent user experience with non-persistent economics

- **Complete application lifecycle management**

With App Volumes, administrators can manage the entire lifecycle of applications, from provisioning, delivery, and maintenance to retirement. This begins with the initial installation of the application to users or devices. When an application requires an update or an upgrade, App Volumes helps speed up this process and additionally supports easy application replacement.

- **Dynamic application delivery to virtualized environments**

App Volumes enables IT administrators to deliver applications and data to any number of virtual machines in seconds. Applications are stored in shared read-only virtual disks that instantly attach to desktops. Applications deployed through App Volumes perform comparably to natively installed Windows applications for end users and provide a seamless desktop experience.

- **Reduced VDI infrastructure costs and improved efficiency**

With App Volumes, enterprises can drive down compute, network, and storage costs by leveraging on-demand delivery of applications in a non-persistent desktop architecture. App Volumes virtual disks can be placed on any supported VMware vSphere datastore. This enables IT to use the most appropriate storage, including fast storage with high read IOPS, such as VMware Virtual SAN™. By combining App Volumes with VMware Horizon®, IT administrators can deliver and manage applications in virtual desktops with greater flexibility, using less storage capacity than they would with Horizon alone.

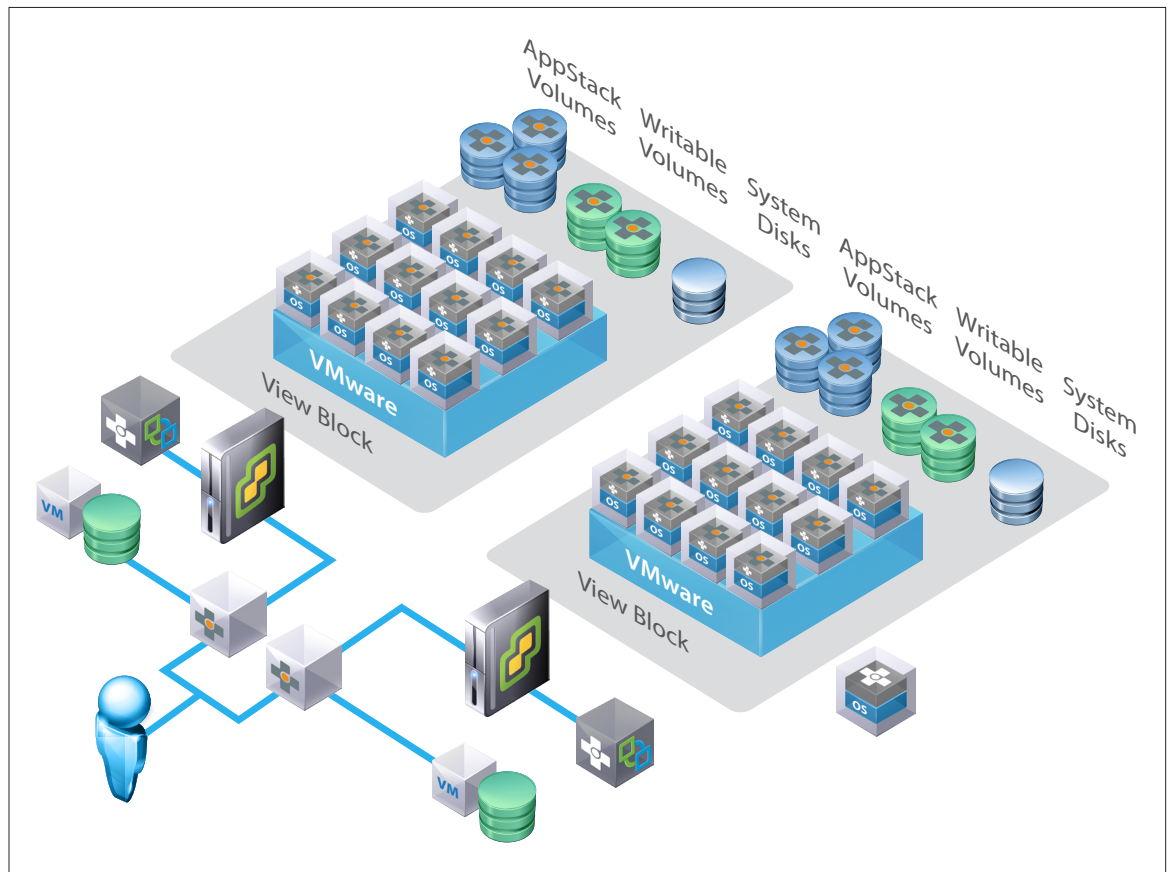
- **Persistent end-user experience across non-persistent environments**

End users can take advantage of fully customized desktops, with the freedom to install their own applications and have them persist across sessions. App Volumes enables a persistent user experience with the cost savings of a non-persistent architecture.

## Architectural Overview

App Volumes integrates a simple agent-server-database architecture into an existing View deployment. 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.

Figure 3 shows the major components of a View environment where App Volumes is deployed.



**Figure 3:** App Volumes Architecture

The major components are described in Table 1.

	<b>App Volumes User</b> – Active Directory (AD) user or organizational unit (OU) account. User account must be assigned local workstation administrator rights.
	<b>App Volumes Manager</b> – A Windows Server system used as the Web Console for administration and configuration of App Volumes and assignment of AppStacks and writable volumes. App Volumes Manager is also used as a broker for the App Volumes agents, for automated assignment of applications and writable volumes during desktop startup and/or user login.
	<b>App Volumes Database</b> – A Microsoft SQL (production) or SQLExpress (non-production) database that contains configuration information for AppStacks, writable volumes, users, machines, entitlements, and transactions.
	<b>App Volumes Agent</b> – Software installed on all Windows desktops where users receive AppStack volumes and writable volume assignment. The agent runs as a service and utilizes a filter driver to handle application calls and file system redirects to AppStack and writable volume VMDKs. Windows desktops do not have to be members of the domain on which the App Volumes Manager server resides.
	<b>AppStack Volume</b> – A read-only volume containing any number of Windows applications. Multiple AppStacks can be mapped to an individual system or user. An individual AppStack can also be mapped to more than one system or user.
	<b>Writable Volume</b> – A user-specific volume where the user is allowed to preserve data. Writable volumes can be used to store user-installed applications and local profile information, such as application settings, application licensing information, and data. A user can be attached to only one writable volume at a time.
	<b>Provisioning Desktop</b> – A clean desktop virtual machine that includes the OS, necessary updates and service packs, and has core applications installed. The provisioning virtual machine must have the App Volumes agent installed and configured to connect to the App Volumes Manager.
	<b>vCenter Server</b> – VMware vCenter Server™ provides centralized management of vSphere virtual infrastructures. App Volumes leverages vCenter for inventory information and operational connectivity to host, virtual machine, and storage resources within a deployed vSphere environment.
	<b>App Volumes View Broker Service</b> – A service installed on a View Connection Server that enables faster user logins for users on View desktops.

**Table 1:** App Volumes Components

## Application Management with AppStacks

App Volumes simplifies virtual desktop infrastructure (VDI), Remote Desktop Session Host (RDSH), and VMware ThinApp® application lifecycle management. The AppStacks real-time application model separates IT-managed applications and application suites into administrator-defined application containers.

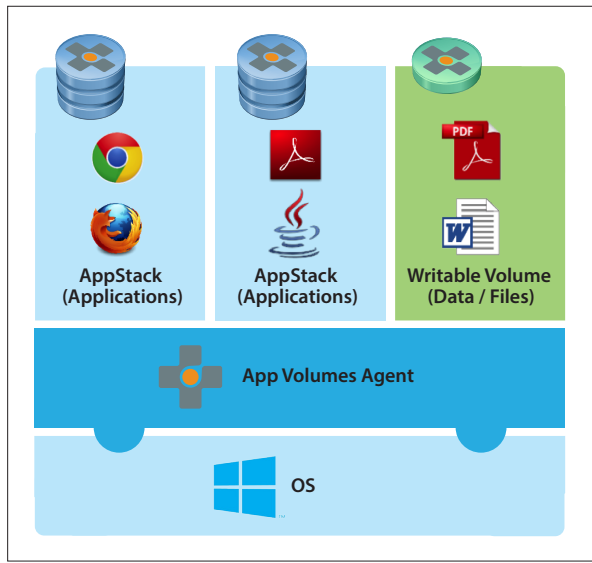


Figure 4: Real-Time App Model

### Application Stacks

AppStacks are read-only volumes containing applications that can be assigned to AD user accounts, groups or 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 to users through AD object assignment. They can make updates available immediately or on next login or reboot.

Administrators can also manage line of business or departmental apps by combining them in an AppStack, which is managed and deployed separately from the core applications.

### Application Suitability

Not every application can or should be installed in an AppStack. The following are key considerations when selecting applications to add to an AppStack:

- Applications running when a user is logged out should be installed in the base image. For this reason, antivirus and security applications should not be provisioned to an AppStack.
- Applications that are dependent on each other should be provisioned to the same AppStack whenever possible.



## ThinApp Integration

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

With App Volumes in a virtual desktop infrastructure, enterprises get the advantages of local deployment modes for ThinApp packages. ThinApp virtual applications can be placed on a shared virtual disk on any available storage that vSphere supports, including local storage, SSD, NAS, or SAN. This arrangement permits thousands of virtual desktops to share a common ThinApp package without the need to stream or copy the package locally.

## RDSH Integration

App Volumes supports AppStack integration with Microsoft RDSH desktops and applications. RDSH desktop integration is as simple as deploying the App Volumes agent on the RDSH desktop image. After the agent is deployed, AppStack assignment is handled through the assignment process (by desktop name) in the App Volumes Manager interface.

Microsoft Windows Server 2008 R2 is the only operating system that supports RDSH integration.

## AppStack Templates

By default, a single 20 GB AppStack template is deployed in an App Volumes environment. This template can be copied and customized depending on how large the AppStack needs to be for a given deployment scenario. See the [App Volumes product documentation](#) for the correct procedures on creating customized AppStack Volume templates.

## Writable Volumes

The Writable Volumes feature is an optional component that enables a per-user volume where the following user-centric data can be installed and configured in different ways and move with the user:

- Local user profile
- Application settings
- Licensing information
- Configuration files
- User-installed apps

The Writable Volumes feature does not provide a user environment management (UEM) solution. Writable volumes complement UEM solutions, which can manage data within writable volumes at a more granular level and provide contextual rules to enforce policy based on different conditions or events. Writable volumes introduce a flexible container used for local user profile delivery across systems.

It is important to note the key differences between AppStacks and writable volumes.

- AppStack VMDKs are mounted as read-only and shared between all desktop virtual machines (VMs) within the data center.
- Writable volumes are dedicated to individual users and mounted as the user authenticates to the desktop. Writable volumes are user-centric and can be assigned to specific computers and then re-assigned to other computers. Writable volumes cannot be shared or reassigned to multiple users.

### Writable Volume Templates

By default, three writable volume templates are deployed when App Volumes Manager is installed:

- User Installed Applications (UIA) Only
- User Profile Only
- UIA and User Profile

These templates can be copied (and resized from their 10 GB default size) as necessary to support individual use-cases or storage scenarios.

### Writable Volume Policies

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

#### Block Login Policy

For a writable volume to work as expected, a user must never have logged in to a computer without the writable volume attached. This is required because a user's existing local profile can interfere with the profile on the 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 to allow or prevent users from logging in.

The **Block Login** option protects users from logging in to persistent desktops without their writable volume. It is needed only for persistent pools. In non-persistent pools, a clean snapshot is provided for each use.

**Limit Attachment Policy**

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

**Defer Create Policy**

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.

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 immediately.

**User Installed Applications**

After UIA or UIA and User Profile writable volumes are provisioned and assigned to a user, that user can install and configure applications. Users require account permissions that allow application installation for this functionality to work properly. App Volumes defers to Microsoft Windows security policies to determine user rights assignments.

**Local User Profiles**

Writable volumes can be configured to capture user profile settings. Windows Local User Profile configurations and application preferences can follow AD users around as they connect to different desktops and types of desktops. This capability does not provide a user environment management capability but does support a local user profile that *roams* between desktop sessions (and even specific desktops). The writable volumes local user profile capability can complement and augment other enterprise UEM solutions.

**Writable Volume Storage Groups**

When defining a writable volume storage group, administrators can set different distribution strategies as needed. A distribution strategy controls how volumes are distributed across the groups of defined datastores.

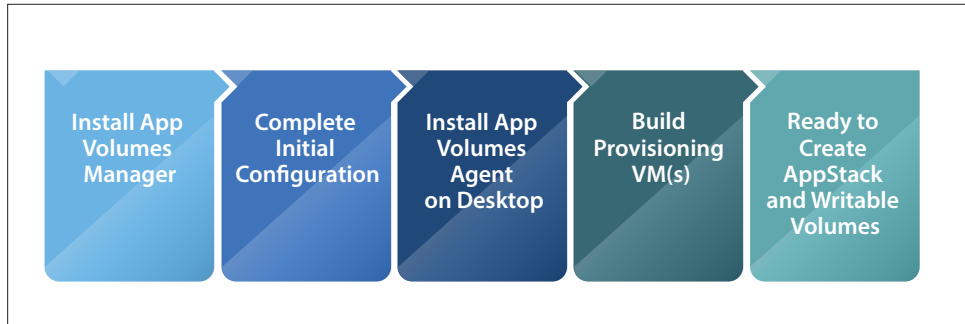
**Spread** – Distribute volumes evenly across all storage locations. When a file is created, the storage with the most available space is selected.

**Spillover** – Distribute volumes by filling each storage location before using the next one. When a writable volume is created, the storage with the least available space is selected.

**Round Robin** – Distribute volumes sequentially by using the storage locations. When a file is created, the datastore with the oldest *last modified* time is selected.

## Deployment Overview

The following diagram shows the basic steps necessary for a typical App Volumes deployment.



**Figure 5:** App Volumes Deployment

Choose a new Windows 2008 Server and make sure 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 the .NET 3.5 framework is deployed.

Create a new database instance for App Volumes. For small environments, a local SQLExpress instance can be deployed upon installation. SQLExpress is recommended only for small, non-production deployments. Configure and use a SQL account with DB\_Owner privileges for installation and operation.

The installer for App Volumes 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.

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 AppStack and writable volumes
- An AD user group for administrator access to App Volumes Manager
- A vCenter instance and administrator-level credentials
- The default storage location for both AppStacks and writable volumes

When the server configuration is complete, deploy the App Volumes agents.

## App Volumes Agent

Deploy agents on desktops that access AppStacks and writable volumes as well as on capture machines (desktops used to capture AppStacks).

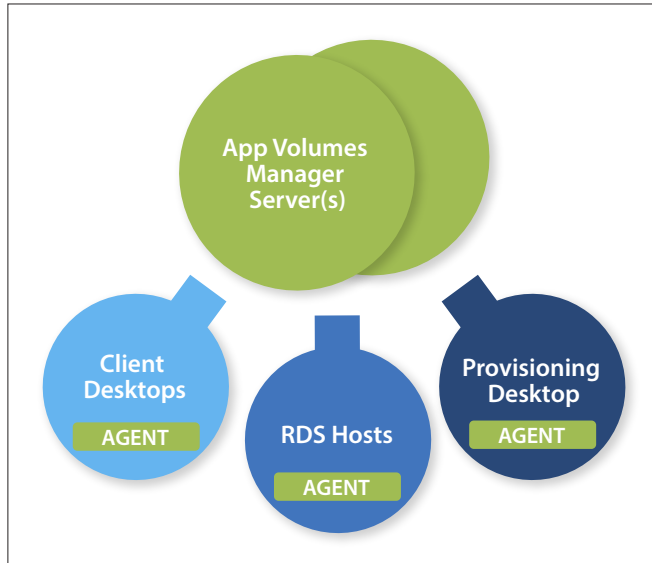


Figure 6: Agent Relationships

## Agent Installation

The App Volumes Agent installation process is the same for clients, RDS hosts, and provisioning VMs. The App Volumes installer is used to deploy the agents. During the installation process, the following information is required:

- Hostname or IP address for the App Volumes Manager – If multiple App Volumes Managers are deployed, 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.

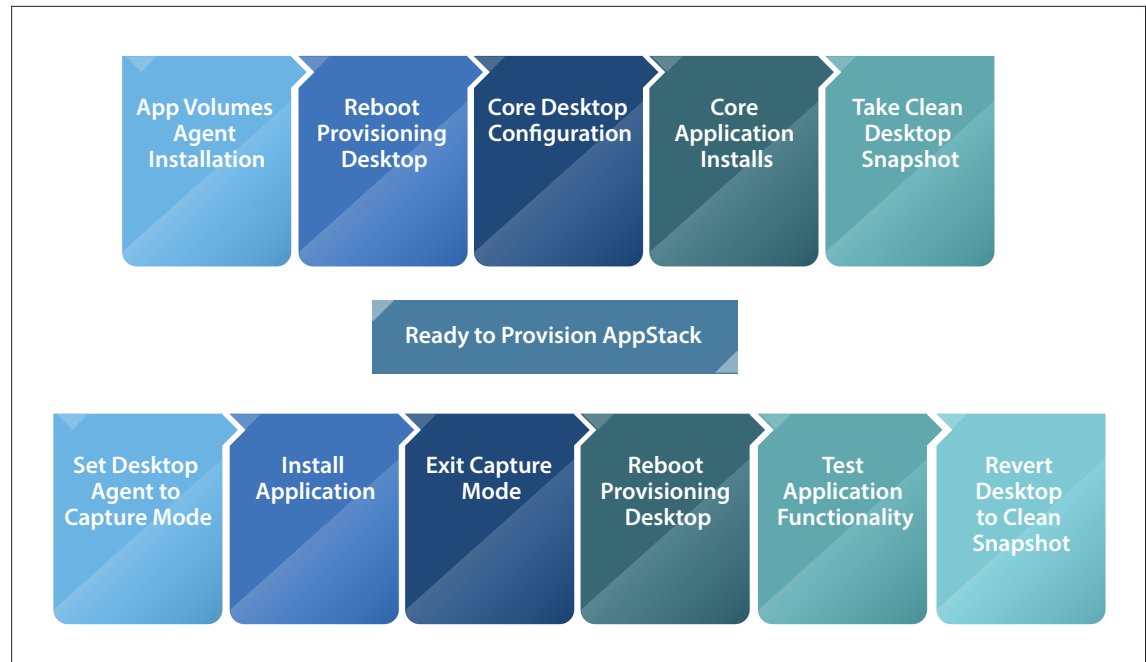
The same installer used for App Volumes Manager is used for agents. Select the **Agent** option after starting the installer.

Enter the hostname/IP address for the App Volumes Manager that the agent should connect to. If multiple App Volumes Managers are deployed, provide the FQDN or IP for the load balancer that provides the front-end for a group of servers.

After the agent is installed, restart the target desktop. If the agent is installed on a provisioning virtual machine, it is essential to take a snapshot of it in its current, clean state.

## Provisioning Desktop Creation

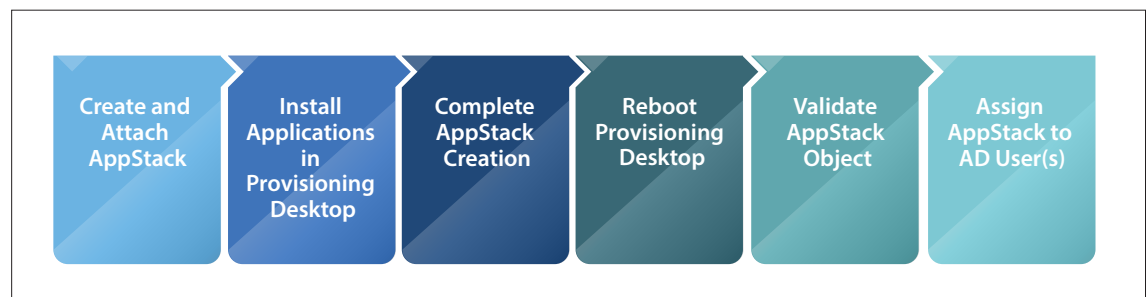
After the App Volumes agent is deployed on a provisioning desktop, administrators should complete any base image configurations and core application installations and take a virtual machine snapshot. This snapshot of the provisioning virtual machine in a *clean* state, before any applications are captured, allows administrators to return the virtual machine to a *ready-to-capture* state.



**Figure 7:** AppStack Creation and Provisioning Workflow

## AppStack Provisioning and Assignment

AppStack provisioning involves placing a clean desktop that is running the App Volumes agent into provisioning status. Only desktops that have the agent installed and are currently connected to the App Volumes Manager server can be selected for AppStack provisioning.



**Figure 8:** AppStack Provisioning

When the provisioning desktop is selected in the App Volumes Manager, a provisioning pop-up message appears. Clear instructions are provided on how to proceed with application installation. Close the App Volumes provisioning pop-up dialog only after application installation is complete, and reboot as necessary.

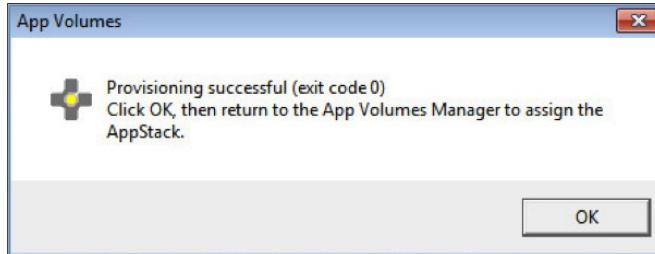


Figure 9: Provisioning Successful

After successful provisioning, **exit code 0** is displayed in the App Volumes agent pop-up window. Return to the App Volumes Manager interface to complete the provisioning process and start testing and assigning the AppStack.

#### Writable Volume Provisioning and Assignment

Writable volume creation is the process of copying a writable volume template and assigning it to a user, a user group, or a computer in AD. Writable volumes are read/write volumes that support user-installed and Windows user profiles. Writable volumes, once created, can be reassigned from one computer to another but cannot be reassigned from one user to another.

#### App Volumes View Broker Service Installation

This optional service should be installed on View Connection Servers. The service initiates the AppStack and writable volume mount process as soon as an entitled user requests a desktop from a View Connection Server. This can reduce the duration of the login process because the AppStack or writable volume mount process is started in the background before the Windows login is completed.

The service is installed using the App Volumes installer. To complete the installation, the FQDN and TCP port of the App Volumes server is specified.

## 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.

REQUIREMENT	DESCRIPTION
AD users	<ul style="list-style-type: none"><li>• An AD account with App Volumes Manager server and vCenter administrator permissions for initial setup and connection</li><li>• A designated AD user group that allows administrator access to the App Volumes Manager server</li><li>• A vCenter administrator-level account for AppStack virtual machine provisioning</li><li>• End users require administrator privileges to install applications into their writable volumes (UIA).</li></ul>
App Volumes Management server	<ul style="list-style-type: none"><li>• Windows Server 2008 R2</li><li>• Application Server Role must be enabled with .NET 3.5 framework deployed</li><li>• 2 vCPUs (recommended 4 vCPUs)</li><li>• 4 GB of RAM</li><li>• 1 GB of disk space</li></ul>
Hypervisor	VMware ESXi™ 5.x with Virtual Center 5.x
App Volumes agent	<ul style="list-style-type: none"><li>• Windows 7</li><li>• 1 GB of RAM</li><li>• 5 MB of disk space</li></ul>
App Volumes Manager database	<ul style="list-style-type: none"><li>• SQL Express 2008 R2 supported for testing/non-production</li><li>• SQL 2008 R2 or 2012 standard and above for production</li><li>• A SQL server account with DB_Owner role is required for installation and operations.</li></ul>
Supported browser to access management console	<ul style="list-style-type: none"><li>• Internet Explorer</li><li>• Firefox</li><li>• Safari</li><li>• Chrome</li></ul>
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 software installation



REQUIREMENT	DESCRIPTION
Basic networking access (defaults – these can be customized)	<ul style="list-style-type: none"> <li>• App Volumes Web Console: 80 (HTTP)/443 (HTTPS)</li> <li>• App Volumes Agent to Server Communications: 80 (HTTP)/443 (HTTPS)</li> <li>• App Volumes Server to vCenter and ESXi Hosts: 443 (SOAP/hostd)</li> <li>• App Volumes Server to remote SQL Server: 1433</li> <li>• App Volumes Server to AD Domain Controller: 389 (LDAP)/636 (LDAPS)</li> </ul>
Storage	<ul style="list-style-type: none"> <li>• 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.</li> <li>• 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 are mounted in favor of duplicate copies that exist on shared storage.</li> </ul>

**Table 2:** App Volumes Deployment Requirements

# 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 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 application(s) to be installed and should also account for application updates, which can be anticipated from time to time. AppStack volumes should always have at least 20 percent free space available to make it easier for administrators to update applications without having to resize the AppStack volumes.

Writable volumes should also be sufficiently sized to accommodate all users' writeable volumes. Storage platforms that allow for volume resizing are helpful if the total number of writable volume users is not known at time of initial solution 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 impacts occur based on user login behavior.
- The option to restrict writable volume access (and thus initial creation) to a certain desktop or group of desktop VMs 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 makes sense to create writable volumes at the time of entitlement, rather than deferring creation.

### Mount on Host

This optional setting 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. **Mount on Host** can decrease the time needed to mount volumes, and it provides resiliency if vCenter is temporarily offline or unavailable.

**Note:** For the **Mount on Host** feature to work, all vSphere hosts must have the same root-level user credentials.

### Mount Local

If local storage is used, AppStacks and writable volumes can be placed on the same datastore where virtual machines reside. The App Volumes policy can be configured to point at local storage before referring to shared storage. If the specified AppStacks and writable volumes are present on the local storage, then these 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 root-level user credentials.

### 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 is providing an application container and brokerage service, storage performance is very important in a production environment. AppStack Volumes are read-only. Depending on utilization patterns, the underlying shared storage platform could be subjected to significant read I/O activity. Consider using flash and hybrid-flash storage technologies for AppStack volumes.

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 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

In a production View environment, it is important to adhere to the sizing and scaling best practices documented in the [View Architecture Planning](#) guide and the vSphere scalability best practices in [Performance Best Practices for vSphere](#). Architectural best practices for vSphere are documented in the [vSphere Resource Management](#) guide.

For non-production 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.

App Volumes Manager servers have been tested with 10,000 active user-to-AppStacks sessions at a time. Adding a load-balanced cluster of App Volumes Managers is a good way to address the additional performance load that comes from an environment that has grown very large.

The current best practices are to limit each virtual desktop to no more than 15 attached AppStack volumes and to connect to no more than 2,000 virtual desktops at a time. In production environments, it is a good practice to combine multiple applications into an AppStack, and to duplicate AppStacks when they are assigned to a large population of users or desktops.

## Security

To support a large production environment, there are some important security configurations that administrators should put in place:

- Open only essential, required firewall ports on App Volumes Manager and SQL Servers. 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.
- Consider using ThinApp to package applications to take advantage of the security benefits of isolation modes, where each ThinApp 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 the desktop operating system. For more information, see [Thin App Resources](#).
- Writable volumes with UIA 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 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, vSphere Replication, and 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.

## Best Practices

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

- Use a production-grade Microsoft SQL instance, rather than the bundled SQL Express instance, that 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.
- 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 work.
- Provision AppStacks on a clean base 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 base image, they should also be in the provisioning virtual machine.
- When reverting a desktop virtual machine that is running the App Volumes agent to a previous snapshot, make sure that the virtual machine is powered off, to avoid syncing issues.
- Enable vCenter datastore browser options. This feature is enabled in the default configuration for vCenter. For more information, see the [vSphere Hardening Guide](#).
- Limit each virtual desktop to no more than 15 attached AppStack volumes.
- If deploying multiple App Volumes Manager servers with a shared SQL database, then after the initial App Volumes Manager deployment, deselect the **Create a new database or overwrite existing database** option during subsequent App Volumes Manager installations.
- When configuring the App Volumes Manager storage policy, select the **Mount on Host** option. This speeds up AppStack and writable volume mount operations and provides resiliency in a situation where vCenter is not online.
- Do not use special characters in naming AppStacks and writable volumes. The following characters cannot be used:  
~!@#\$%^&(){}[]|,`#\$/:\*?<>' "&
- 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 for provisioning before, revert that virtual machine to the clean snapshot before provisioning a new AppStack.
- 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 [App Volumes product documentation](#).
- 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.
- When there is an application conflict, the last AppStack wins. Use the **Override Precedence** option to define AppStack ordering in the Directory tab, which lists AppStacks assigned to a user, computer, or group.
- Applications that should be available to the OS when no user is logged in should be placed in the base image.
- When deploying user-assigned AppStacks, use fewer AppStacks rather than more. You can include more than one application per AppStack.

## About the Author and Contributors

Tristan Todd, End-User-Computing Architect, VMware, wrote this paper.

The following individuals provided content and review:

- Stephane Asselin, Senior End-User-Computing Architect, Technical Enablement, End-User Computing, VMware
- Dean Flaming, Senior Technical Marketing Manager, Lighthouse Program and Technical Enablement, End-User Computing, VMware

To comment on this paper, contact the VMware End-User-Computing Technical Marketing team at [euc\\_tech\\_content\\_feedback@vmware.com](mailto:euc_tech_content_feedback@vmware.com).

## References

- [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](#)

