TECHNICAL WHITE PAPER – **AUGUST 2017**

VMWARE APP VOLUMES 2.X DATABASE BEST PRACTICES

VMware App Volumes 2.x

vmware®

Table of Contents

Introduction

Often when an application performs poorly or stops working altogether, the problem can be traced back to a database issue—the type and edition used, whether the host system has adequate resources, whether the database is sized properly, whether the data and log files are being managed properly, or whether high-availability strategies are being used effectively, if at all. In the case of the VMware App Volumes™ database, even though the database is relatively small, proper setup and maintenance are crucial. Every App Volumes operation is scheduled using the database, and each operation requires multiple SQL queries.

In this guide, we have gathered all the pertinent App Volumes 2.x database best practices together and organized them into the following sections:

- Database sizing
- Database performance
- High availability

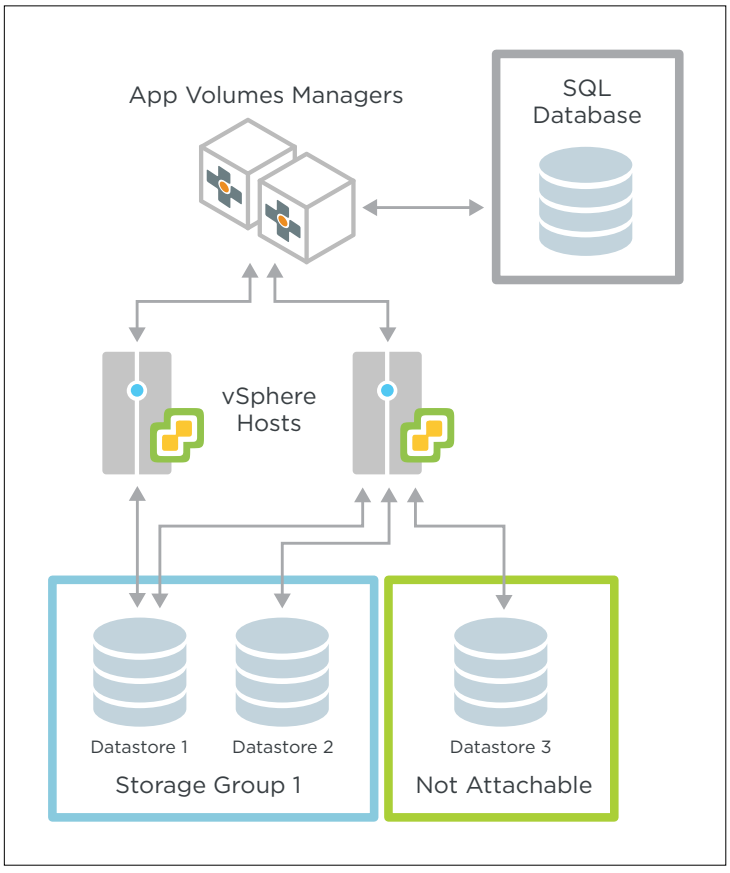


Figure 1: Example Logical Architecture of an App Volumes Deployment

Database Sizing

The following types of objects are contained in an App Volumes database:

- **Static configuration information** – These settings, which are usually configured during initial setup and rarely change over the lifetime of the deployment, include App Volumes machine manager configuration, Active Directory (AD) configuration, and storage group configuration. These settings do not usually consume more than 5 MB.
- **Environmental information** – Information about environmental objects—including VM registration and state, VMware ESXi™ hosts, datastores, writable volumes `.zip` files, Active Directory users and groups, and domain controllers—is stored in the database permanently, even if the underlying physical object is removed.

  For example, when a user logs in to a VM that has the App Volumes Agent, information about the user is pulled from AD and stored in the App Volumes database. If the user is removed from AD, although the user is automatically hidden from the App Volumes Manager console, the user information is not deleted from the App Volumes database.
- **Information about assignments, AppStacks, and writable volumes** – Configuration information about AppStacks, AppStack assignments, applications in AppStacks, writable volumes, and `.vmdk` and `.vhd` files is stored in the App Volumes database until the underlying configuration is removed.
- **Auditing information** – Activity logs and system messages are retained indefinitely or until cleared manually. The number of records created depends on how the environment is used and on how often the configuration is changed. Activity logs include events such as computer startup and shutdown, user login and logout, administrator activity, and AD synchronization.
- **Dynamic data** – Information about administrator sessions, pending tasks, and delayed jobs is stored in the database temporarily to coordinate work between multiple App Volumes Manager servers. To estimate the size of the transaction log, assume that dynamic data requires the same amount of space as static data plus 20 percent.

Sizing Example 1

In this example, we have 1,000 users who log in once a day to access a single desktop and have one writable volume and two AppStacks assigned per user. VMs are distributed across 20 ESXi hosts, which use the AppStacks **mount local** option. We can use the following calculations for database sizing:

~5 MB of static configuration

plus

- 1 KB per VM * 1,000 = 1,000 KB
- 1 KB per ESXi host * 20 = 20 KB
- 4 KB per AD user account * 1,000 = 4,000 KB
- 4 KB per AD computer account * 1,000 = 4,000 KB
- 3 KB per datastore * 20 = 60 KB
- 1 KB per domain controller * 2 = 2 KB

~9 MB of environmental information

plus

- 16 KB per AppStack * 2 = 32 KB
- 6 KB per assignment * 1,000 = 6,000 KB
- 2 KB per application * 20 = 40 KB
- 22 KB per writable volume * 1,000 = 22,000 KB
- 5 KB per writable `.vmdk` file * 1,000 = 5,000 KB
- 5 KB per replicated AppStack `.vmdk` file * 21 = 105 KB

~33 MB for assignments, AppStacks, and writable volumes

This totals ~47 MB for configuration.

+20% of dynamic data

This gives us the approximate size of ~56 MB for 1,000 users.

Additionally, auditing information will require:

0.5 KB per login/logout operation * (2+2+6) * 1,000 = 10,000 KB = ~5 MB/day = 1,825 MB/year

Note: The login/logout operations are composed of 1 pre-startup event + 1 startup event + 1 login event + 1 logout event + 3 attach-volume events + 3 detach-volume events.

0.5 KB per sync operation * (1,000 users + 1,000 computers) = 1,000 KB = ~1 MB/day = 365 MB/year

=~2 GB per year for 1,000 users

Based on these calculations, we could safely set the following size for the database files:

- 2 GB for the primary ROWS data file
- 10 MB for the transaction log file if the simple recovery model is in use

Sizing Example 2

In this example, we have 1,000 users who log in twice a day to access a single desktop and have one writable volume and four AppStacks assigned per user. VMs are distributed across 20 ESXi hosts, which use the AppStacks **mount local** option. We can use the following calculations for database sizing:

~5 MB of static configuration

plus

- 1 KB per VM * 1,000 = 1,000 KB
- 1 KB per ESXi host * 20 = 20 KB
- 4 KB per AD user account * 1,000 = 4,000 KB
- 4 KB per AD computer account * 1,000 = 4,000 KB
- 3 KB per datastore * 20 = 60 KB
- 1 KB per domain controller * 2 = 2 KB

~9 MB of environmental information

plus

- 16 KB per AppStack * 4 = 64 KB
- 6 KB per assignment * 1,000 = 6,000 KB
- 2 KB per application * 20 = 40 KB
- 22 KB per writable volume * 1,000 = 22,000 KB
- 5 KB per writable `.vmdk` file * 1,000 = 5,000 KB
- 5 KB per replicated AppStack `.vmdk` file * 21 = 105 KB

~33 MB for assignments, AppStacks, and writable volumes

This totals ~47 MB for configuration.

+20% of dynamic data

This gives us the approximate size of ~56 MB for 1,000 users.

Additionally, auditing information will require:

0.5 KB per login/logout operation * (2+4+20) * 1,000 = 13,000 KB = ~13 MB/day = 4,745 MB/year

Note: The login/logout operations are composed of 1 pre-startup event + 1 startup event + 2 login events + 2 logout events + 10 attach-volume events + 10 detach-volume events.

0.5 KB per sync operation * (1,000 users + 1,000 computers) = 1,000 KB = ~1 MB/day = 365 MB/year

=~5 GB per year for 1,000 users

Based on these calculations, we could safely set the following size for the database files:

- 5.1 GB for the database file
- 10 MB for the transaction log file if the simple recovery model is in use

Sizing Example 3

In this example, we have 5,000 users who log in twice a day to access a single desktop and have one writable volume and five AppStacks assigned per user. VMs are distributed across 100 ESXi hosts, which use the AppStacks **mount local** option. We can use the following calculations for database sizing:

~5 MB of static configuration

plus

- 1 KB per VM * 5,000 = 5,000 KB
- 1 KB per ESXi host * 100 = 100 KB
- 4 KB per AD user account * 5,000 = 20,000 KB
- 4 KB per AD computer account * 5,000 = 20,000 KB
- 3 KB per datastore * 100 = 300 KB
- 1 KB per domain controller * 2 = 2 KB

~46 MB of environmental information

- 16 KB per AppStack * 5 = 80 KB
- 6 KB per assignment * 5,000 = 30,000 KB
- 2 KB per application * 100 = 200 KB
- 22 KB per writable volume * 5,000 = 110,000 KB
- 5 KB per writable `.vmdk` file * 5,000 = 25,000 KB
- 5 KB per replicated AppStack `.vmdk` file * 101 = 505 KB

~ 166 MB for assignments, AppStacks, and writable volumes

This totals ~217 MB for configuration.

+20% of dynamic data

This gives us the approximate size of ~261 MB for 1,000 users.

Additionally, auditing information will require:

0.5 KB per login/logout operation * (2+4+24) * 5,000 = 75,000 KB = ~75 MB/day = 27,375 MB/year

Note: The login/logout operations are composed of 1 pre-startup event + 1 startup event + 2 login events + 2 logout events + 12 attach-volume events + 12 detach-volume events.

0.5 KB per sync operation * (5,000 users + 5,000 computers) = 5,000 KB = ~5 MB/day = 1,825 MB/year

=~29 GB per year for 5,000 users

Based on these calculations, we could safely set the following size for the database files:

- 30 GB for the database file
- 10 MB for the transaction log file if the simple recovery model is in use

Important: These examples provide a high-level estimate and are intended to give you a good idea of the approximate size of the database. The actual size will vary based on the way SQL Server stores data.

Sizing of the transaction log with a different database recovery model (such as full or bulk) is much less precise and much more dependent on the environment.

Database Performance

Microsoft SQL Server is a high-resource-consuming application. Despite the relatively small size of the App Volumes database and the lack of critical customer data in it, availability of the database is crucial for App Volumes Manager performance. All operations are scheduled using the database, and all operations require multiple SQL queries.

Use the following guidelines for best performance:

- For production App Volumes environments, use an Enterprise or Standard edition of Microsoft SQL Server. Do not use SQL Server Express.
- When designing the SQL Server environment that supports App Volumes, be sure to follow Microsoft best practices. SQL Server limits, not App Volumes limits, apply to the number of objects per database.
- Place SQL Server on a dedicated VM that has adequate CPUs, RAM, and disk space to support the SQL instance. See the VMware knowledge base article Tips for configuring Microsoft SQL Server in a virtual machine (1002951).
- With regard to transaction logs, VMware testing shows that when SQL Server is configured to autogrow the transaction log, all transactions are delayed or stalled, causing an increase in response times. VMware recommends that if you use the full recovery model, set the size of the transaction log large enough so that the autogrow option is used only as a contingency for unexpected growth, or set the transaction log to a fixed size.

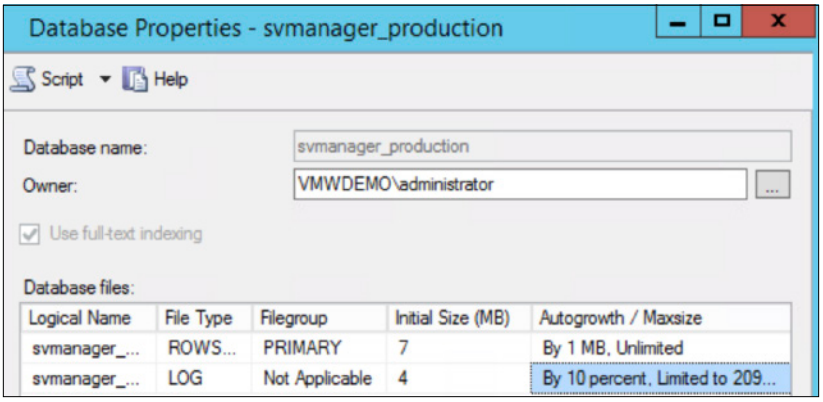


Figure 2: Transaction Log Set to a Fixed Maximum Size

In this case, you should set up an SQL Alert so that when the transaction log reaches 50 percent full, the transaction log is backed up, thus freeing it. This strategy maintains the transaction log at a reasonable size without impacting SQL Server performance.

Note: SQL Server Agent must be enabled to send alerts.

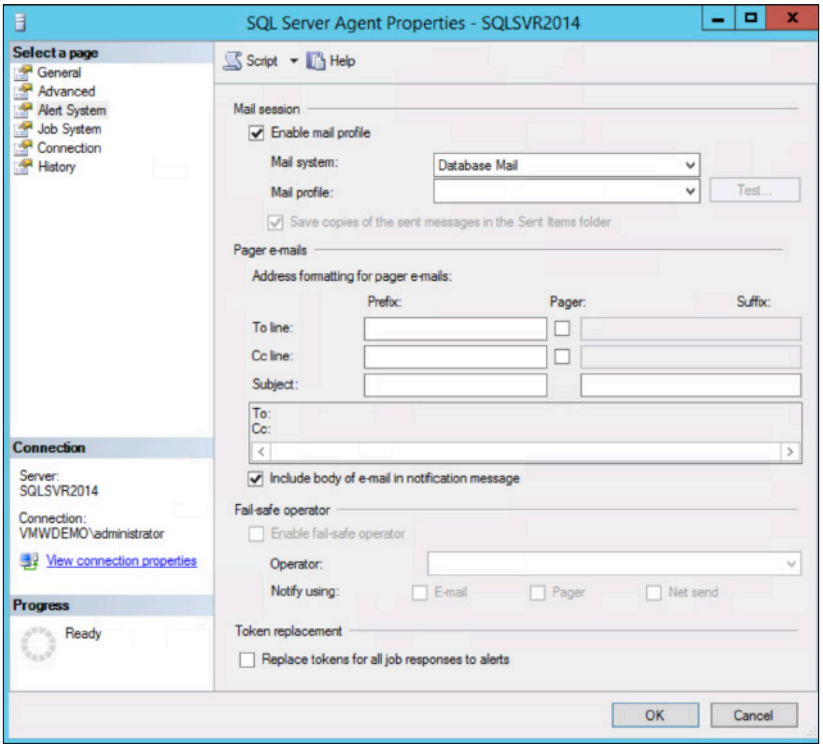


Figure 3: Enabling SQL Server Agent to Send Alerts

- If auditing data is not required, consider pruning the VMware App Volumes SQL database. To perform this operation, see the VMware knowledge base article Pruning the VMware App Volumes SQL database (2132454).
- Use the correct configuration and number of App Volumes Manager servers for the number of end users. In most environments, multiple App Volumes Manager servers are deployed. For better performance and reliability, make the following changes:

APP VOLUMES ITEM	FOR <1,000 USERS	FOR >1,000 TO <5,000 USERS	FOR >5,000 USERS
Number of App Volumes Manager servers	2	3	3+1 for each 2,500 users
Number of CPUs on each App Volumes server	4	6	8
RAM on each App Volumes server	4	8	16
Changes to `Manager\config\database.yml`	pool:35 checkout_timeout:45	pool:20 checkout_timeout:45	pool:20 checkout_timeout:45
Changes to `Manager\clock.yml` (section `medium:`)	workers: 1	servers: 6 workers: 1	servers: 8 workers: 1

Table 1: Configuring App Volumes for Various Numbers of Users

High Availability

High-availability solutions for Microsoft SQL Server include:

- **Always On availability groups** – VMware recommends this method because it guards against disk failure and provides constant availability of the database.
- **Database mirroring** – This method is supported when it is configured with a witness server. Note that Microsoft has deprecated this feature in favor of Always On availability groups.
- **Always On failover cluster instances** – Microsoft recommends this method if you plan to use data protection through a third-party shared disk solution (a SAN) rather than through SQL Server.

For more information, see the Microsoft SQL Server documentation for High Availability Solutions (SQL Server).

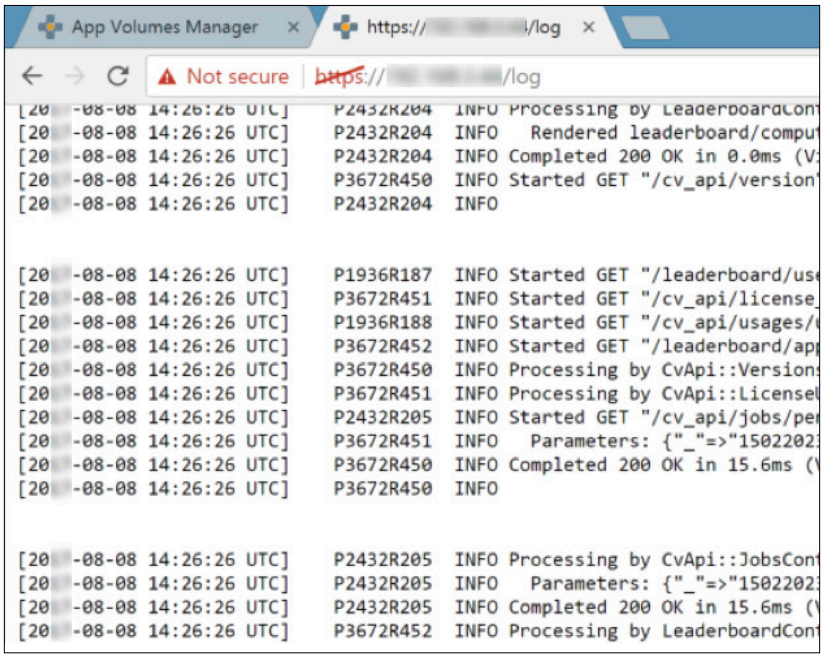
Configuring App Volumes Manager to Use a Highly Available Database

This document does not provide instructions for installing and configuring SQL Server to use SQL Server mirroring or Always On availability groups. For instructions for SQL Server–side configuration, see the Microsoft SQL Server documentation.

The **SQL Server Mirroring** and **Always On availability groups** options for the App Volumes database can be configured only after the SQL database is created when you install the first instance of App Volumes Manager. Also, manual configuration is required after upgrading App Volumes Manager.

Follow these steps to configure each App Volumes Manager server to use a highly available database:

1. Configure the highly available database by following the Microsoft SQL Server documentation.
2. On the App Volumes Manager server, install the SQL Server native client (64-bit version) that matches the version of the SQL Server.
3. Using the ODBC control panel, configure the new system DSN to use the SQL Server native client and point to a primary and a failover SQL server.
4. Use a text editor to open the file `c:\Program Files (x86)\CloudVolumes\Manager\config\database.yml`.
5. Change the line `dsn: svmanager` to use the name of the DSN configured in Step 3.
6. Save your changes to the `database.yml` file.
7. Reboot the App Volumes Manager server.
8. Verify log access to the App Volumes Manager UI.
   a. In a new browser tab, go to https://<Server_Address>/log.
   b. Verify that log entries appear on the page, as shown in the following figure.



Database Maintenance

The **SQL Server Mirroring** and **Always On availability groups** options require that the database use the full recovery model. The transaction log must be backed up to prevent excessive growth and fragmentation.

VMware recommends configuring an SQL Server alert to monitor the following SQL performance counter:

SQLServer:Databases - Percent Log Used – appvolumes_database

For more information, see the Microsoft SQL Server documents Monitor and Respond to Events and Implement Jobs.

Deploying App Volumes in Multi-Site Environments

Many companies have a presence in multiple geographic locations. In scenarios that require App Volumes to be deployed in multiple locations across the globe, VMware recommends separate App Volumes deployments.

App Volumes Manager requires a reliable and constant connection to the SQL database. Any delays or loss of communication between App Volumes Manager and its SQL database will cause performance and stability issues. These issues include but are not limited to

- Slower user logins and logouts
- Delays in AppStack attachment
- Duplicate jobs executed by multiple managers

VMware recommends that an App Volumes deployment not span data centers with a network latency that can affect communications between App Volumes Manager and SQL Server. Also consider the history of communications reliability and past performance of the connection.

In cases where the latency between App Volumes Manager and SQL Server is higher than 15 ms, use a separate App Volumes deployment. A pod architecture in which all components are local to the enclosure/server rack—including App Volumes Manager and SQL Server—provides the best results.

Appendix A: Disk Space Requirements for the App Volumes Database

The following table summarizes the requirements for the various types of objects in the App Volumes database.

STATIC DATA	TOTAL SIZE
Configuration information for App Volumes machine managers, AD, and storage groups	5 MB
ENVIRONMENTAL OBJECTS	**SIZE FOR EACH RECORD**
VM registration and VM state information	1 KB
ESXi host	1 KB
AD User, group, computer, or organizational unit	4 KB
VMware vSphere® datastore	3 KB
Domain controller (discovered)	1 KB
`.zip` file updates for writable volumes	Size of the `.zip` file + 30%
ASSIGNMENTS, APPSTACKS, AND WRITABLE VOLUMES	
AppStack	16 KB
AppStack assignment	6 KB
Application in the AppStack	2 KB
Writable volume	22 KB
`.vmdk` or `.vhd` file (including AppStacks replicated across storage groups)	5 KB
AUDITING INFORMATION	
Activity log events	0.5–1 KB per operation
System messages such as agent errors, communication failures	0.5–2 KB per message
DYNAMIC DATA	
Administrator sessions	1 KB per session
Pending tasks and delayed jobs	1 KB per job

Table 2: : Approximate Size for Each Record in the App Volumes Database

Additional Resources

For more information, you can explore the following resources:

- VMware App Volumes
- VMware App Volumes Documentation
- VMware Knowledge Base
- VMware Product Evaluation
- VMware Product Guide
- VMware Product Interoperability Matrixes
- White papers
- Self-help resources
- VMware consultation and support
  - VMware Horizon Support Center
  - VMware Consulting Professional Services Organization

About the Authors

The following team wrote this document:

- Denis Gundarev, Application Solutions Architect, App Volumes R&D, VMware
- Stéphane Asselin, Lead Product Architect, End-User Computing, VMware

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

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com