Understand NAS file access

ONTAP 9

NetApp
January 11, 2023

Table of Contents

Understand NAS file access

Namespaces and junction points

Namespaces and junction points overview

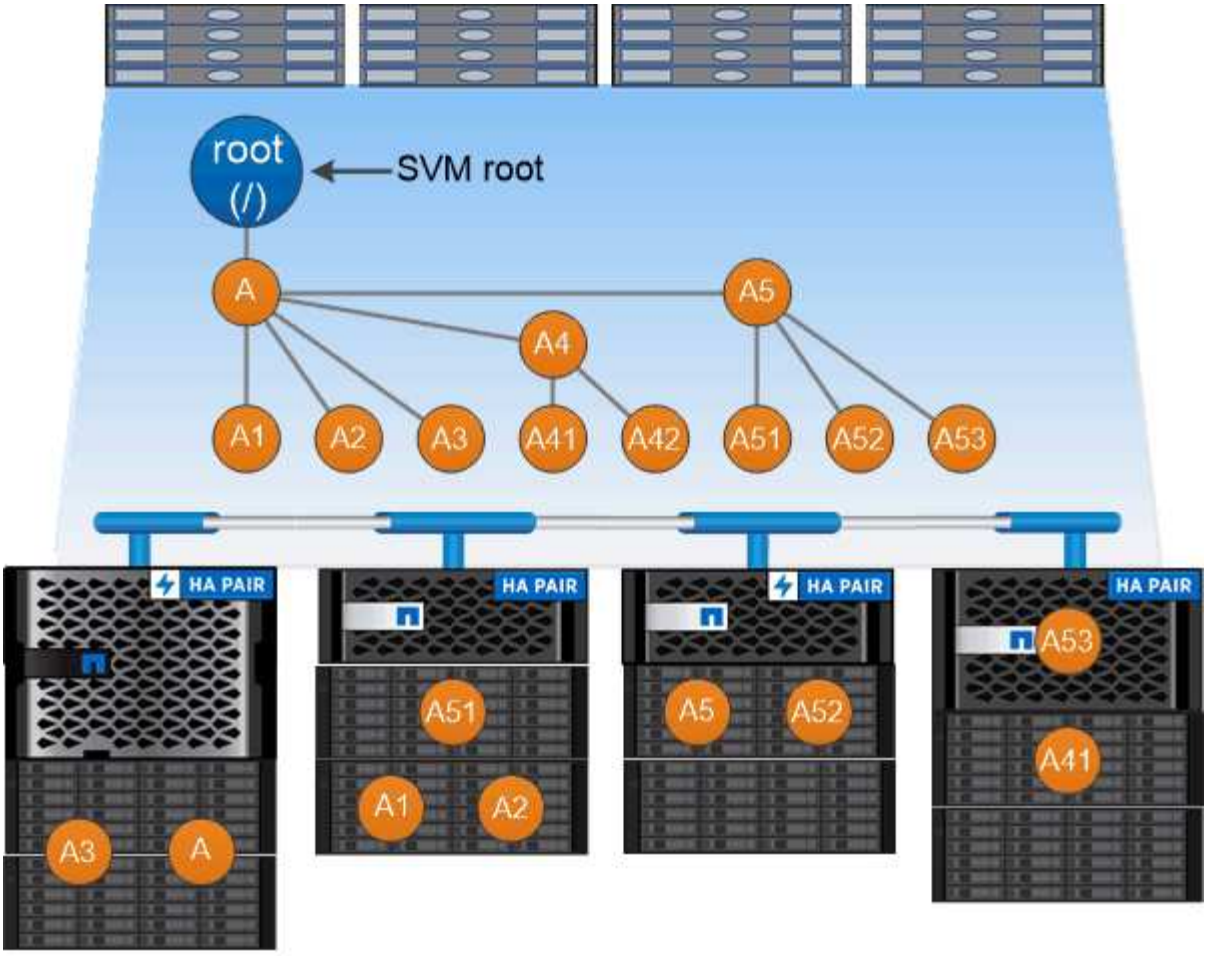
A NAS *namespace* is a logical grouping of volumes joined together at *junction points* to create a single file system hierarchy. A client with sufficient permissions can access files in the namespace without specifying the location of the files in storage. Junctioned volumes can reside anywhere in the cluster.

Rather than mounting every volume containing a file of interest, NAS clients mount an NFS *export* or access an SMB *share.* The export or share represents the entire namespace or an intermediate location within the namespace. The client accesses only the volumes mounted below its access point.

You can add volumes to the namespace as needed. You can create junction points directly below a parent volume junction or on a directory within a volume. A path to a volume junction for a volume named “vol3” might be `/vol1/vol2/vol3`, or `/vol1/dir2/vol3`, or even `/dir1/dir2/vol3`. The path is called the *junction path.*

Every SVM has a unique namespace. The SVM root volume is the entry point to the namespace hierarchy.

To ensure that data remains available in the event of a node outage or failover, you should create a *load-sharing mirror* copy for the SVM root volume.



A namespace is a logical grouping of volumes joined together at junction points to create a single file system hierarchy.

Example

The following example creates a volume named “home4” located on SVM vs1 that has a junction path `/eng/home`:

```
cluster1::> volume create -vserver vs1 -volume home4 -aggregate aggr1
-size 1g -junction-path /eng/home
[Job 1642] Job succeeded: Successful
```

What the typical NAS namespace architectures are

There are several typical NAS namespace architectures that you can use as you create your SVM name space. You can choose the namespace architecture that matches your business and workflow needs.

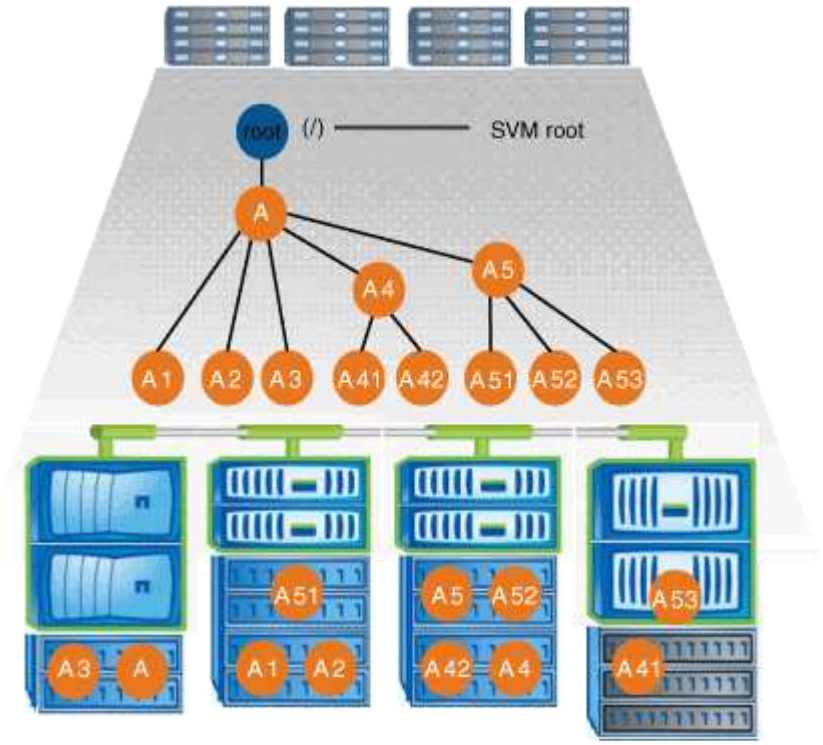
The top of the namespace is always the root volume, which is represented by a slash (/). The namespace architecture under the root falls into three basic categories:

- A single branched tree, with only a single junction to the root of the namespace

- Multiple branched trees, with multiple junction points to the root of the namespace
- Multiple stand-alone volumes, each with a separate junction point to the root of the name space

Namespace with single branched tree

An architecture with a single branched tree has a single insertion point to the root of the SVM namespace. The single insertion point can be either a junctioned volume or a directory beneath the root. All other volumes are mounted at junction points beneath the single insertion point (which can be a volume or a directory).

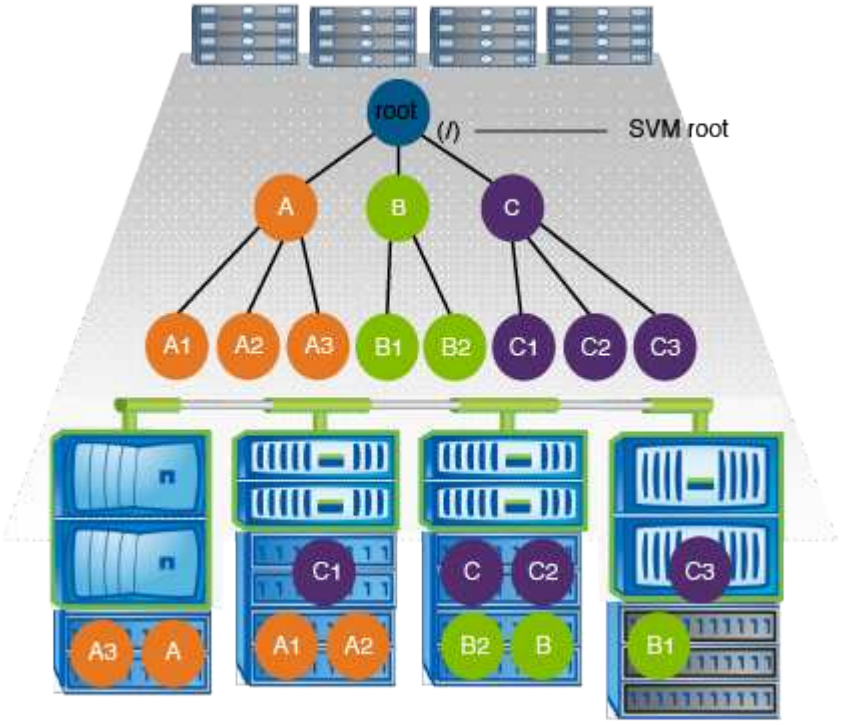


For example, a typical volume junction configuration with the above namespace architecture might look like the following configuration, where all volumes are junctioned below the single insertion point, which is a directory named “data”:

```
                    Junction                        Junction
Vserver Volume      Active   Junction Path          Path Source
------- ----------- -------- --------------------   -----------
vs1     corp1       true     /data/dir1/corp1       RW_volume
vs1     corp2       true     /data/dir1/corp2       RW_volume
vs1     data1       true     /data/data1            RW_volume
vs1     eng1        true     /data/data1/eng1       RW_volume
vs1     eng2        true     /data/data1/eng2       RW_volume
vs1     sales       true     /data/data1/sales      RW_volume
vs1     vol1        true     /data/vol1             RW_volume
vs1     vol2        true     /data/vol2             RW_volume
vs1     vol3        true     /data/vol3             RW_volume
vs1     vs1_root    -        /                      -
```

Namespace with multiple branched trees

An architecture with multiple branched trees has multiple insertion points to the root of the SVM namespace. The insertion points can be either junctioned volumes or directories beneath the root. All other volumes are mounted at junction points beneath the insertion points (which can be volumes or directories).



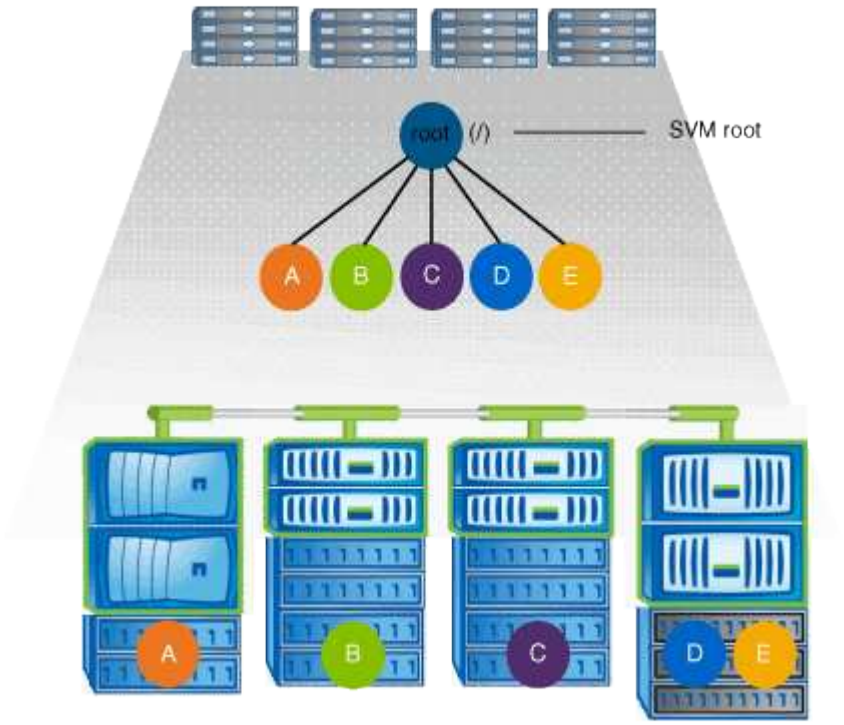
For example, a typical volume junction configuration with the above namespace architecture might look like the following configuration, where there are three insertion points to the root volume of the SVM. Two insertion points are directories named “data” and “projects”. One insertion point is a junctioned volume named “audit”:

```
                    Junction                        Junction
Vserver Volume      Active   Junction Path          Path Source
------- ----------- -------- --------------------   -----------
vs1     audit       true     /audit                 RW_volume
vs1     audit_logs1 true     /audit/logs1           RW_volume
vs1     audit_logs2 true     /audit/logs2           RW_volume
vs1     audit_logs3 true     /audit/logs3           RW_volume
vs1     eng         true     /data/eng              RW_volume
vs1     mktg1       true     /data/mktg1            RW_volume
vs1     mktg2       true     /data/mktg2            RW_volume
vs1     project1    true     /projects/project1     RW_volume
vs1     project2    true     /projects/project2     RW_volume
vs1     vs1_root    -        /                      -
```

Namespace with multiple stand-alone volumes

In an architecture with stand-alone volumes, every volume has an insertion point to the root of the SVM namespace; however, the volume is not junctioned below another volume. Each volume has a unique path,

and is either junctioned directly below the root or is junctioned under a directory below the root.



For example, a typical volume junction configuration with the above namespace architecture might look like the following configuration, where there are five insertion points to the root volume of the SVM, with each insertion point representing a path to one volume.

```
                    Junction                          Junction
Vserver Volume      Active   Junction Path            Path Source
------- ----------- -------- -------------------      -----------
vs1     eng         true     /eng                     RW_volume
vs1     mktg        true     /vol/mktg                RW_volume
vs1     project1    true     /project1                RW_volume
vs1     project2    true     /project2                RW_volume
vs1     sales       true     /sales                   RW_volume
vs1     vs1_root    -        /                        -
```

How ONTAP controls access to files

How ONTAP controls access to files overview

ONTAP controls access to files according to the authentication-based and file-based restrictions that you specify.

When a client connects to the storage system to access files, ONTAP has to perform two tasks:

- Authentication

ONTAP has to authenticate the client by verifying the identity with a trusted source. In addition, the authentication type of the client is one method that can be used to determine whether a client can access data when configuring export policies (optional for CIFS).

- Authorization

  ONTAP has to authorize the user by comparing the user's credentials with the permissions configured on the file or directory and determining what type of access, if any, to provide.

To properly manage file access control, ONTAP must communicate with external services such as NIS, LDAP, and Active Directory servers. Configuring a storage system for file access using CIFS or NFS requires setting up the appropriate services depending on your environment in ONTAP.

Authentication-based restrictions

With authentication-based restrictions, you can specify which client machines and which users can connect to the storage virtual machine (SVM).

ONTAP supports Kerberos authentication from both UNIX and Windows servers.

File-based restrictions

ONTAP evaluates three levels of security to determine whether an entity is authorized to perform a requested action on files and directories residing on an SVM. Access is determined by the effective permissions after evaluation of the three security levels.

Any storage object can contain up to three types of security layers:

- Export (NFS) and share (SMB) security

  Export and share security applies to client access to a given NFS export or SMB share. Users with administrative privileges can manage export and share-level security from SMB and NFS clients.

- Storage-Level Access Guard file and directory security

  Storage-Level Access Guard security applies to SMB and NFS client access to SVM volumes. Only NTFS access permissions are supported. For ONTAP to perform security checks on UNIX users for access to data on volumes for which Storage-Level Access Guard has been applied, the UNIX user must map to a Windows user on the SVM that owns the volume.

  > If you view the security settings on a file or directory from an NFS or SMB client, you will not see Storage-Level Access Guard security. Storage-Level Access Guard security cannot be revoked from a client, even by a system (Windows or UNIX) administrator.

- NTFS, UNIX, and NFSv4 native file-level security

  Native file-level security exists on the file or directory that represents the storage object. You can set file-level security from a client. File permissions are effective regardless of whether SMB or NFS is used to access the data.

How ONTAP handles NFS client authentication

How ONTAP handles NFS client authentication overview

NFS clients must be properly authenticated before they can access data on the SVM. ONTAP authenticates the clients by checking their UNIX credentials against the name services that you configure.

When an NFS client connects to the SVM, ONTAP obtains the UNIX credentials for the user by checking different name services, depending on the name services configuration of the SVM. ONTAP can check credentials for local UNIX accounts, NIS domains, and LDAP domains. At least one of them must be configured so that ONTAP can successfully authenticate the user. You can specify multiple name services and the order in which ONTAP searches them.

In a pure NFS environment with UNIX volume security styles, this configuration is sufficient to authenticate and provide the proper file access for a user connecting from an NFS client.

If you are using mixed, NTFS, or unified volume security styles, ONTAP must obtain a SMB user name for the UNIX user for authentication with a Windows domain controller. This can happen either by mapping individual users using local UNIX accounts or LDAP domains, or by using a default SMB user instead. You can specify which name services ONTAP searches in which order, or specify a default SMB user.

How ONTAP uses name services

ONTAP uses name services to obtain information about users and clients. ONTAP uses this information to authenticate users accessing data on or administering the storage system, and to map user credentials in a mixed environment.

When you configure the storage system, you must specify what name services you want ONTAP to use for obtaining user credentials for authentication. ONTAP supports the following name services:

- Local users (file)
- External NIS domains (NIS)
- External LDAP domains (LDAP)

You use the `vserver services name-service ns-switch` command family to configure SVMs with the sources to search for network information and the order in which to search them. These commands provide the equivalent functionality of the `/etc/nsswitch.conf` file on UNIX systems.

When an NFS client connects to the SVM, ONTAP checks the specified name services to obtain the UNIX credentials for the user. If name services are configured correctly and ONTAP can obtain the UNIX credentials, ONTAP successfully authenticates the user.

In an environment with mixed security styles, ONTAP might have to map user credentials. You must configure name services appropriately for your environment to allow ONTAP to properly map user credentials.

ONTAP also uses name services for authenticating SVM administrator accounts. You must keep this in mind when configuring or modifying the name service switch to avoid accidentally disabling authentication for SVM administrator accounts. For more information about SVM administration users, see xref:./nfs-admin/../authentication/index.html[Administrator authentication and RBAC.

How ONTAP grants SMB file access from NFS clients

ONTAP uses Windows NT File System (NTFS) security semantics to determine whether a UNIX user, on an NFS client, has access to a file with NTFS permissions.

ONTAP does this by converting the user's UNIX User ID (UID) into a SMB credential, and then using the SMB credential to verify that the user has access rights to the file. A SMB credential consists of a primary Security Identifier (SID), usually the user's Windows user name, and one or more group SIDs that correspond to Windows groups of which the user is a member.

The time ONTAP takes converting the UNIX UID into a SMB credential can be from tens of milliseconds to hundreds of milliseconds because the process involves contacting a domain controller. ONTAP maps the UID to the SMB credential and enters the mapping in a credential cache to reduce the verification time caused by the conversion.

How the NFS credential cache works

When an NFS user requests access to NFS exports on the storage system, ONTAP must retrieve the user credentials either from external name servers or from local files to authenticate the user. ONTAP then stores these credentials in an internal credential cache for later reference. Understanding how the NFS credential caches works enables you to handle potential performance and access issues.

Without the credential cache, ONTAP would have to query name services every time an NFS user requested access. On a busy storage system that is accessed by many users, this can quickly lead to serious performance problems, causing unwanted delays or even denials to NFS client access.

With the credential cache, ONTAP retrieves the user credentials and then stores them for a predetermined amount of time for quick and easy access should the NFS client send another request. This method offers the following advantages:

- It eases the load on the storage system by handling fewer requests to external name servers (such as NIS or LDAP).
- It eases the load on external name servers by sending fewer requests to them.
- It speeds up user access by eliminating the wait time for obtaining credentials from external sources before the user can be authenticated.

ONTAP stores both positive and negative credentials in the credential cache. Positive credentials means that the user was authenticated and granted access. Negative credentials means that the user was not authenticated and was denied access.

By default, ONTAP stores positive credentials for 24 hours; that is, after initially authenticating a user, ONTAP uses the cached credentials for any access requests by that user for 24 hours. If the user requests access after 24 hours, the cycle starts over: ONTAP discards the cached credentials and obtains the credentials again from the appropriate name service source. If the credentials changed on the name server during the previous 24 hours, ONTAP caches the updated credentials for use for the next 24 hours.

By default, ONTAP stores negative credentials for two hours; that is, after initially denying access to a user, ONTAP continues to deny any access requests by that user for two hours. If the user requests access after 2 hours, the cycle starts over: ONTAP obtains the credentials again from the appropriate name service source. If the credentials changed on the name server during the previous two hours, ONTAP caches the updated credentials for use for the next two hours.

Copyright information

Trademark information