NAS protocols

NetApp Solutions

NetApp
August 02, 2023

Table of Contents

NAS protocols

NAS protocols overview

Previous: Firewall.

NAS protocols include NFS (v3 and v4.1) and SMB/CIFS (2.x and 3.x). These protocols are how CVS allows shared access to data across multiple NAS clients. In addition, Cloud Volumes Service can provide access to NFS and SMB/CIFS clients simultaneously (dual-protocol) while honoring all of the identity and permission settings on files and folders in the NAS shares. To maintain the highest possible data transfer security, Cloud Volumes Service supports protocol encryption in flight using SMB encryption and NFS Kerberos 5p.

Dual-protocol is available with CVS-Performance only.

Next: Basics of NAS protocols.

Basics of NAS protocols

Previous: NAS protocols overview.

NAS protocols are ways for multiple clients on a network to access the same data on a storage system, such as Cloud Volumes Service on GCP. NFS and SMB are the defined NAS protocols and operate on a client/server basis where Cloud Volumes Service acts as the server. Clients send access, read, and write requests to the server, and the server is responsible for coordinating the locking mechanisms for files, storing permissions and handling identity and authentication requests.

For example, the following general process is followed if a NAS client wants to create a new file in a folder.

1. The client asks the server for information about the directory (permissions, owner, group, file ID, available space, and so on); the server responds with the information if the requesting client and user have the necessary permissions on the parent folder.
2. If the permissions on the directory allow access, the client then asks the server if the file name being created already exists in the file system. If the file name is already in use, creation fails. If the file name does not exist, the server lets the client know it can proceed.
3. The client issues a call to the server to create the file with the directory handle and file name and sets the access and modified times. The server issues a unique file ID to the file to make sure that no other files are created with the same file ID.
4. The client sends a call to check file attributes before the WRITE operation. If permissions allow it, the client then writes the new file. If locking is used by the protocol/application, the client asks the server for a lock to prevent other clients from accessing the file while locked to prevent data corruption.

Next: NFS.

NFS

Previous: Basics of NAS protocols_overview.

NFS is a distributed file system protocol that is an open IETF standard defined in Request for Comments (RFC) that allows anyone to implement the protocol.

Volumes in Cloud Volumes Service are shared out to NFS clients by exporting a path that is accessible to a client or set of clients. Permissions to mount these exports are defined by export policies and rules, which are configurable by Cloud Volumes Service administrators.

The NetApp NFS implementation is considered a gold standard for the protocol and is used in countless enterprise NAS environments. The following sections cover NFS and specific security features available in Cloud Volumes Service and how they are implemented.

Default local UNIX users and groups

Cloud Volumes Service contains several default UNIX users and groups for various basic functionalities. These users and groups cannot currently be modified or deleted. New local users and groups cannot currently be added to Cloud Volumes Service. UNIX users and groups outside of the default users and groups need to be provided by an external LDAP name service.

The following table shows the default users and groups and their corresponding numeric IDs. NetApp recommends not creating new users or groups in LDAP or on the local clients that re-use these numeric IDs.

Default users: numeric IDs	Default groups: numeric IDs
• root:0 • pcuser:65534 • nobody:65535	• root:0 • daemon:1 • pcuser:65534 • nobody:65535

When using NFSv4.1, the root user might display as nobody when running directory listing commands on NFS clients. This is due to the client's ID domain mapping configuration. See the section called NFSv4.1 and the nobody user/group for details on this issue and how to resolve it.

The root user

In Linux, the root account has access to all commands, files, and folders in a Linux-based file system. Because of the power of this account, security best practices often require the root user to be disabled or restricted in some fashion. In NFS exports, the power a root user has over the files and folders can be controlled in Cloud Volumes Service through export policies and rules and a concept known as root squash.

Root squashing ensures that the root user accessing an NFS mount is squashed to the anonymous numeric user 65534 (see the section "The anonymous user") and is currently only available when using CVS-Performance by selecting Off for root access during export policy rule creation. If the root user is squashed to the anonymous user, it no longer has access to run chown or setuid/setgid commands (the sticky bit) on files or folders in the NFS mount, and files or folders created by the root user show the anon UID as the owner/group. In addition, NFSv4 ACLs cannot be modified by the root user. However, the root user still has access to chmod and deleted files that it does not have explicit permissions for. If you want to limit access to a root user's file

and folder permissions, consider using a volume with NTFS ACLs, creating a Windows user named `root`, and applying the desired permissions to the files or folders.

The anonymous user

The anonymous (anon) user ID specifies a UNIX user ID or username that is mapped to client requests that arrive without valid NFS credentials. This can include the root user when root squashing is used. The anon user in Cloud Volumes Service is 65534.

This UID is normally associated with the username `nobody` or `nfsnobody` in Linux environments. Cloud Volumes Service also uses 65534 as the local UNIX user` pcuser` (see the section "Default local UNIX users and groups"), which is also the default fallback user for Windows to UNIX name mappings when no valid matching UNIX user can be found in LDAP.

Because of the differences in usernames across Linux and Cloud Volumes Service for UID 65534, the name string for users mapped to 65534 might not match when using NFSv4.1. As a result, you might see `nobody` as the user on some files and folders. See the section "NFSv4.1 and the nobody user/group" for information about this issue and how to resolve it.

Access control/exports

Initial export/share access for NFS mounts is controlled through host- based export policy rules contained within an export policy. A host IP, host name, subnet, netgroup, or domain is defined to allow access to mount the NFS share and the level of access allowed to the host. Export policy rule configuration options depend on the Cloud Volumes Service level.

For CVS-SW, the following options are available for export-policy configuration:

- **Client match.** Comma-separated list of IP addresses, comma-separated list of hostnames, subnets, netgroups, domain names.
- **RO/RW access rules.** Select read/write or read only to control level of access to export.CVS-Performance provides the following options:
- **Client match.** Comma-separated list of IP addresses, comma-separated list of hostnames, subnets, netgroups, domain names.
- **RO/RW access rules.** Select read/write or read only to control level of access to export.
- **Root access (on/off).** Configures root squash (see the section "The root user" for details).
- **Protocol type.** This limits access to the NFS mount to a specific protocol version. When specifying both NFSv3 and NFSv4.1 for the volume, either leave both blank or check both boxes.
- **Kerberos security level (when Enable Kerberos is selected).** Provides the options of krb5, krb5i, and/or krb5p for read-only or read-write access.

Change ownership (chown) and change group (chgrp)

NFS on Cloud Volumes Service only allows the root user to run chown/chgrp on files and folders. Other users see an `Operation not permitted` error— even on files they own. If you use root squash (as covered in the section "The root user"), the root is squashed to a nonroot user and is not allowed access to chown and chgrp. There are currently no workarounds in Cloud Volumes Service to allow chown and chgrp for non-root users. If ownership changes are required, consider using dual protocol volumes and set the security style to NTFS to control permissions from the Windows side.

Permission management

Cloud Volumes Service supports both mode bits (such as 644, 777, and so on for rwx) and NFSv4.1 ACLs to control permissions on NFS clients for volumes that use the UNIX security style. Standard permission management is used for these (such as chmod, chown, or nfs4_setfacl) and work with any Linux client that supports them.

Additionally, when using dual protocol volumes set to NTFS, NFS clients can leverage Cloud Volumes Service name mapping to Windows users, which then are used to resolve the NTFS permissions. This requires an LDAP connection to Cloud Volumes Service to provide numeric-ID-to- username translations because Cloud Volumes Service requires a valid UNIX username to map properly to a Windows username.

Providing granular ACLs for NFSv3

Mode bit permissions cover only owner, group, and everyone else in the semantics—meaning that there are no granular user access controls in place for basic NFSv3. Cloud Volumes Service does not support POSIX ACLs, nor extended attributes (such as chattr), so granular ACLs are only possible in the following scenarios with NFSv3:

- NTFS security style volumes (CIFS server required) with valid UNIX to Windows user mappings.
- NFSv4.1 ACLs applied using an admin client mounting NFSv4.1 to apply ACLs.

Both methods require an LDAP connection for UNIX identity management and a valid UNIX user and group information populated (see the section “LDAP”) and are only available with CVS-Performance instances. To use NTFS security style volumes with NFS, you must use dual-protocol (SMB and NFSv3) or dual-protocol (SMB and NFSv4.1), even if no SMB connections are made. To use NFSv4.1 ACLs with NFSv3 mounts, you must select `Both (NFSv3/NFSv4.1)` as the protocol type.

Regular UNIX mode bits don’t provide the same level of granularity in permissions that NTFS or NFSv4.x ACLs provide. The following table compares the permission granularity between NFSv3 mode bits and NFSv4.1 ACLs. For information about NFSv4.1 ACLs, see nfs4_acl - NFSv4 Access Control Lists.

NFSv3 mode bits	NFSv4.1 ACLs
• Set user ID on execution • Set group ID on execution • Save swapped text (not defined in POSIX) • Read permission for owner • Write permission for owner • Execute permission for owner on a file; or look up (search) permission for owner in directory • Read permission for group • Write permission for group • Execute permission for group on a file; or look up (search) permission for group in directory • Read permission for others • Write permission for others • Execute permission for others on a file; or look up (search) permission for others in directory	Access control entry (ACE) types (Allow/Deny/Audit) * Inheritance flags * directory-inherit * file-inherit * no-propagate-inherit * inherit-only Permissions * read-data (files) / list-directory (directories) * write-data (files) / create-file (directories) * append-data (files) / create-subdirectory (directories) * execute (files) / change-directory (directories) * delete * delete-child * read-attributes * write-attributes * read-named-attributes * write-named-attributes * read-ACL * write-ACL * write-owner * Synchronize

Finally, NFS group membership (in both NFSv3 and NFSV4.x) is limited to a default maximum of 16 for AUTH_SYS as per the RPC packet limits. NFS Kerberos provides up to 32 groups and NFSv4 ACLs remove the limitation by way of granular user and group ACLs (up to 1024 entries per ACE).

Additionally, Cloud Volumes Service provides extended group support to extend the maximum supported groups up to 32. This requires an LDAP connection to an LDAP server that contains valid UNIX user and group identities. For more information about configuring this, see Creating and managing NFS volumes in the Google documentation.

NFSv3 user and group IDs

NFSv3 user and group IDs come across the wire as numeric IDs rather than names. Cloud Volumes Service does no username resolution for these numeric IDs with NFSv3, with UNIX security style volumes using just mode bits. When NFSv4.1 ACLs are present, a numeric ID lookup and/or name string lookup is needed to resolve the ACL properly—even when using NFSv3. With NTFS security style volumes, Cloud Volumes Service must resolve a numeric ID to a valid UNIX user and then map to a valid Windows user to negotiate access rights.

Security limitations of NFSv3 user and group IDs

With NFSv3, the client and server never have to confirm that the user attempting a read or write with a numeric ID is a valid user; it is just implicitly trusted. This opens the file system up to potential breaches simply by spoofing any numeric ID. To prevent security holes like this, there are a few options available to Cloud Volumes Service.

- Implementing Kerberos for NFS forces users to authenticate with a username and password or keytab file to get a Kerberos ticket to allow access into a mount. Kerberos is available with CVS-Performance instances and only with NFSv4.1.

- Limiting the list of hosts in your export policy rules limits which NFSv3 clients have access to the Cloud Volumes Service volume.
- Using dual-protocol volumes and applying NTFS ACLs to the volume forces NFSv3 clients to resolve numeric IDs to valid UNIX usernames to authenticate properly to access mounts. This requires enabling LDAP and configuring UNIX user and group identities.
- Squashing the root user limits the damage a root user can do to an NFS mount but does not completely remove risk. For more information, see the section "The root user."

Ultimately, NFS security is limited to what the protocol version you are using offers. NFSv3, while more performant in general than NFSv4.1, does not provide the same level of security.

NFSv4.1

NFSv4.1 provides greater security and reliability as compared to NFSv3, for the following reasons:

- Integrated locking through a lease-based mechanism
- Stateful sessions
- All NFS functionality over a single port (2049)
- TCP only
- ID domain mapping
- Kerberos integration (NFSv3 can use Kerberos, but only for NFS, not for ancillary protocols such as NLM)

NFSv4.1 dependencies

Because of the additionally security features in NFSv4.1, there are some external dependencies involved that were not needed to use NFSv3 (similar to how SMB requires dependencies such as Active Directory).

NFSv4.1 ACLs

Cloud Volumes Service offers support for NFSv4.x ACLs, which deliver distinct advantages over normal POSIX-style permissions, such as the following:

- Granular control of user access to files and directories
- Better NFS security
- Improved interoperability with CIFS/SMB
- Removal of the NFS limitation of 16 groups per user with AUTH_SYS security
- ACLs bypass the need for group ID (GID) resolution, which effectively removes the GID limitNFSv4.1 ACLs are controlled from NFS clients—not from Cloud Volumes Service. To use NFSv4.1 ACLs, be sure your client's software version supports them and the proper NFS utilities are installed.

Compatibility between NFSv4.1 ACLs and SMB clients

NFSv4 ACLs are different from Windows file-level ACLs (NTFS ACLs) but carry similar functionality. However, in multiprotocol NAS environments, if NFSv4.1 ACLs are present and you are using dual-protocol access (NFS and SMB on the same datasets), clients using SMB2.0 and later won't be able to view or manage ACLs from Windows security tabs.

How NFSv4.1 ACLs work

For reference, the following terms are defined:

- **Access control list (ACL).** A list of permissions entries.
- **Access control entry (ACE).** A permission entry in the list.

When a client sets an NFSv4.1 ACL on a file during a SETATTR operation, Cloud Volumes Service sets that ACL on the object, replacing any existing ACL. If there is no ACL on a file, then the mode permissions on the file are calculated from OWNER@, GROUP@, and EVERYONE@. If there are any existing SUID/SGID/STICKY bits on the file, they are not affected.

When a client gets an NFSv4.1 ACL on a file during the course of a GETATTR operation, Cloud Volumes Service reads the NFSv4.1 ACL associated with the object, constructs a list of ACEs, and returns the list to the client. If the file has an NT ACL or mode bits, then an ACL is constructed from mode bits and is returned to the client.

Access is denied if a DENY ACE is present in the ACL; access is granted if an ALLOW ACE exists. However, access is also denied if neither of the ACEs is present in the ACL.

A security descriptor consists of a security ACL (SACL) and a discretionary ACL (DACL). When NFSv4.1 interoperates with CIFS/SMB, the DACL is one-to-one mapped with NFSv4 and CIFS. The DACL consists of the ALLOW and the DENY ACEs.

If a basic `chmod` is run on a file or folder with NFSv4.1 ACLs set, existing user and group ACLs are preserved, but the default OWNER@, GROUP@, EVERYONE@ ACLs are modified.

A client using NFSv4.1 ACLs can set and view ACLs for files and directories on the system. When a new file or subdirectory is created in a directory that has an ACL, that object inherits all ACEs in the ACL that have been tagged with the appropriate inheritance flags.

If a file or directory has an NFSv4.1 ACL, that ACL is used to control access no matter which protocol is used to access the file or directory.

Files and directories inherit ACEs from NFSv4 ACLs on parent directories (possibly with appropriate modifications) as long as the ACEs have been tagged with the correct inheritance flags.

When a file or directory is created as the result of an NFSv4 request, the ACL on the resulting file or directory depends on whether the file creation request includes an ACL or only standard UNIX file access permissions. The ACL also depends on whether the parent directory has an ACL.

- If the request includes an ACL, that ACL is used.
- If the request includes only standard UNIX file access permissions and the parent directory does not have an ACL, the client file mode is used to set standard UNIX file access permissions.
- If the request includes only standard UNIX file access permissions and the parent directory has a noninheritable ACL, a default ACL based on the mode bits passed into the request is set on the new object.
- If the request includes only standard UNIX file access permissions but the parent directory has an ACL, the ACEs in the parent directory's ACL are inherited by the new file or directory as long as the ACEs have been tagged with the appropriate inheritance flags.

ACE permissions

NFSv4.1 ACLs permissions uses a series of upper- and lower-case letter values (such as `rxtncy`) to control access. For more information about these letter values, see HOW TO: Use NFSv4 ACL.

NFSv4.1 ACL behavior with umask and ACL inheritance

NFSv4 ACLs provide the ability to offer ACL inheritance. ACL inheritance means that files or folders created beneath objects with NFSv4.1 ACLs set can inherit the ACLs based on the configuration of the ACL inheritance flag.

Umask is used to control the permission level at which files and folders are created in a directory without administrator interaction. By default, Cloud Volumes Service allows umask to override inherited ACLs, which is expected behavior as per RFC 5661.

ACL formatting

NFSv4.1 ACLs have specific formatting. The following example is an ACE set on a file:

```
A::ldapuser@domain.netapp.com:rwatTnNcCy
```

The preceding example follows the ACL format guidelines of:

```
type:flags:principal:permissions
```

A type of `A` means “allow.” The inherit flags are not set in this case, because the principal is not a group and does not include inheritance. Also, because the ACE is not an AUDIT entry, there is no need to set the audit flags. For more information about NFSv4.1 ACLs, see http://linux.die.net/man/5/nfs4_acl.

If the NFSv4.1 ACL is not set properly (or a name string cannot be resolved by the client and server), the ACL might not behave as expected, or the ACL change might fail to apply and throw an error.

Sample errors include:

```
Failed setxattr operation: Invalid argument
Scanning ACE string 'A:: user@rwaDxtTnNcCy' failed.
```

Explicit DENY

NFSv4.1 permissions can include explicit DENY attributes for OWNER, GROUP, and EVERYONE. That is because NFSv4.1 ACLs are default-deny, which means that if an ACL is not explicitly granted by an ACE, then it is denied. Explicit DENY attributes override any ACCESS ACEs, explicit or not.

DENY ACEs are set with an attribute tag of `D`.

In the example below, GROUP@ is allowed all read and execute permissions, but denied all write access.

```
sh-4.1$ nfs4_getfacl /mixed
A::ldapuser@domain.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A:g:GROUP@:rxtncy
D:g:GROUP@:waDTC
A::EVERYONE@:rxtncy
D::EVERYONE@:waDTC
```

DENY ACEs should be avoided whenever possible because they can be confusing and complicated; ALLOW ACLs that are not explicitly defined are implicitly denied. When DENY ACEs are set, users might be denied access when they expect to be granted access.

The preceding set of ACEs is equivalent to 755 in mode bits, which means:

- The owner has full rights.
- Groups have read only.
- Others have read only.

However, even if permissions are adjusted to the 775 equivalent, access can be denied because of the explicit DENY set on EVERYONE.

NFSv4.1 ID domain mapping dependencies

NFSv4.1 leverages ID domain mapping logic as a security layer to help verify that a user attempting access to an NFSv4.1 mount is indeed who they claim to be. In these cases, the username and group name coming from the NFSv4.1 client appends a name string and sends it to the Cloud Volumes Service instance. If that username/group name and ID string combination does not match, then the user and/or group is squashed to the default nobody user specified in the `/etc/idmapd.conf` file on the client.

This ID string is a requirement for proper permission adherence, especially when NFSv4.1 ACLs and/or Kerberos are in use. As a result, name service server dependencies such as LDAP servers are necessary to ensure consistency across clients and Cloud Volumes Service for proper user and group name identity resolution.

Cloud Volumes Service uses a static default ID domain name value of `defaultv4iddomain.com`. NFS clients default to the DNS domain name for its ID domain name settings, but you can manually adjust the ID domain name in `/etc/idmapd.conf`.

If LDAP is enabled in Cloud Volumes Service, then Cloud Volumes Service automates the NFS ID domain to change to what is configured for the search domain in DNS and clients won't need to be modified unless they use different DNS domain search names.

When Cloud Volumes Service can resolve a username or group name in local files or LDAP, the domain string is used and non-matching domain IDs squash to nobody. If Cloud Volumes Service cannot find a username or group name in local files or LDAP, the numeric ID value is used and the NFS client resolves the name properly (this is similar to NFSv3 behavior).

Without changing the client's NFSv4.1 ID domain to match what the Cloud Volumes Service volume is using, you see the following behavior:

- UNIX users and groups with local entries in Cloud Volumes Service (such as root, as defined in local UNIX users and groups) are squashed to the nobody value.
- UNIX users and groups with entries in LDAP (if Cloud Volumes Service is configured to use LDAP) squashes to nobody if DNS domains are different between NFS clients and Cloud Volumes Service.
- UNIX users and groups with no local entries or LDAP entries use the numeric ID value and resolve to the name specified on the NFS client. If no name exists on the client, only the numeric ID is shown.

The following shows the results of the preceding scenario:

```
# ls -la /mnt/home/prof1/nfs4/
total 8
drwxr-xr-x 2 nobody nobody 4096 Feb  3 12:07 .
drwxrwxrwx 7 root   root   4096 Feb  3 12:06 ..
-rw-r--r-- 1   9835   9835    0 Feb  3 12:07 client-user-no-name
-rw-r--r-- 1 nobody nobody    0 Feb  3 12:07 ldap-user-file
-rw-r--r-- 1 nobody nobody    0 Feb  3 12:06 root-user-file
```

When the client and server ID domains match, this is how the same file listing looks:

```
# ls -la
total 8
drwxr-xr-x 2 root   root          4096 Feb  3 12:07 .
drwxrwxrwx 7 root   root          4096 Feb  3 12:06 ..
-rw-r--r-- 1   9835          9835    0 Feb  3 12:07 client-user-no-name
-rw-r--r-- 1 apache apache-group    0 Feb  3 12:07 ldap-user-file
-rw-r--r-- 1 root   root            0 Feb  3 12:06 root-user-file
```

For more information about this issue and how to resolve it, see the section "NFSv4.1 and the nobody user/group."

Kerberos dependencies

If you plan to use Kerberos with NFS, you must have the following with Cloud Volumes Service:

- Active Directory domain for Kerberos Distribution Center services (KDC)
- Active Directory domain with user and group attributes populated with UNIX information for LDAP functionality (NFS Kerberos in Cloud Volumes Service requires a user SPN to UNIX user mapping for proper functionality.)
- LDAP enabled on the Cloud Volumes Service instance
- Active Directory domain for DNS services

NFSv4.1 and the nobody user/group

One of the most common issues seen with an NFSv4.1 configuration is when a file or folder is shown in a listing using `ls` as being owned by the `user:group` combination of `nobody:nobody`.

For example:

```
sh-4.2$ ls -la | grep prof1-file
-rw-r--r-- 1 nobody nobody    0 Apr 24 13:25 prof1-file
```

And the numeric ID is `99`.

```
sh-4.2$ ls -lan | grep prof1-file
-rw-r--r-- 1 99 99    0 Apr 24 13:25 prof1-file
```

In some instances, the file might show the correct owner but `nobody` as the group.

```
sh-4.2$ ls -la | grep newfile1
-rw-r--r-- 1 prof1  nobody    0 Oct  9  2019 newfile1
```

Who is nobody?

The `nobody` user in NFSv4.1 is different from the `nfsnobody` user. You can view how an NFS client sees each user by running the `id` command:

```
# id nobody
uid=99(nobody) gid=99(nobody) groups=99(nobody)
# id nfsnobody
uid=65534(nfsnobody) gid=65534(nfsnobody) groups=65534(nfsnobody)
```

With NFSv4.1, the `nobody` user is the default user defined by the `idmapd.conf` file and can be defined as any user you want to use.

```
# cat /etc/idmapd.conf | grep nobody
#Nobody-User = nobody
#Nobody-Group = nobody
```

Why does this happen?

Because security through name string mapping is a key tenet of NFSv4.1 operations, the default behavior when a name string does not match properly is to squash that user to one that won't normally have any access to files and folders owned by users and groups.

When you see `nobody` for the user and/or group in file listings, this generally means something in NFSv4.1 is misconfigured. Case sensitivity can come into play here.

For example, if user1@CVSDEMO.LOCAL (uid 1234, gid 1234) is accessing an export, then Cloud Volumes Service must be able to find user1@CVSDEMO.LOCAL (uid 1234, gid 1234). If the user in Cloud Volumes Service is USER1@CVSDEMO.LOCAL, then it won't match (uppercase USER1 versus lowercase user1). In

many cases, you can see the following in the messages file on the client:

```
May 19 13:14:29 centos7 nfsidmap[17481]: nss_getpwnam: name
'root@defaultv4iddomain.com' does not map into domain 'CVSDEMO.LOCAL'
May 19 13:15:05 centos7 nfsidmap[17534]: nss_getpwnam: name 'nobody' does
not map into domain 'CVSDEMO.LOCAL'
```

The client and server must both agree that a user is indeed who they are claiming to be, so you must check the following to ensure that the user that the client sees has the same information as the user that Cloud Volumes Service sees.

- **NFSv4.x ID domain.** Client: `idmapd.conf` file; Cloud Volumes Service uses `defaultv4iddomain.com` and cannot be changed manually. If using LDAP with NFSv4.1, Cloud Volumes Service changes the ID domain to what the DNS search domain is using, which is the same as the AD domain.
- **User name and numeric IDs.** This determines where the client is looking for user names and leverages the name service switch configuration—client: `nsswitch.conf` and/or local passwd and group files; Cloud Volumes Service does not allow modifications to this but automatically adds LDAP to the configuration when it is enabled.
- **Group name and numeric IDs.** This determines where the client is looking for group names and leverages the name service switch configuration—client: `nsswitch.conf` and/or local passwd and group files; Cloud Volumes Service does not allow modifications to this but automatically adds LDAP to the configuration when it is enabled.

In almost all cases, if you see `nobody` in user and group listings from clients, the issue is user or group name domain ID translation between Cloud Volumes Service and the NFS client. To avoid this scenario, use LDAP to resolve user and group information between clients and Cloud Volumes Service.

Viewing name ID strings for NFSv4.1 on clients

If you are using NFSv4.1, there is a name-string mapping that takes place during NFS operations, as previously described.

In addition to using `/var/log/messages` to find an issue with NFSv4 IDs, you can use the nfsidmap -l command on the NFS client to view which usernames have properly mapped to the NFSv4 domain.

For example, this is output of the command after a user that can be found by the client and Cloud Volumes Service accesses an NFSv4.x mount:

```
# nfsidmap -l
4 .id_resolver keys found:
  gid:daemon@CVSDEMO.LOCAL
  uid:nfs4@CVSDEMO.LOCAL
  gid:root@CVSDEMO.LOCAL
  uid:root@CVSDEMO.LOCAL
```

When a user that does not map properly into the NFSv4.1 ID domain (in this case, `netapp-user`) tries to access the same mount and touches a file, they are assigned `nobody:nobody`, as expected.

```
# su netapp-user
sh-4.2$ id
uid=482600012(netapp-user), 2000(secondary)
sh-4.2$ cd /mnt/nfs4/
sh-4.2$ touch newfile
sh-4.2$ ls -la
total 16
drwxrwxrwx  5 root   root   4096 Jan 14 17:13 .
drwxr-xr-x. 8 root   root     81 Jan 14 10:02 ..
-rw-r--r--  1 nobody nobody    0 Jan 14 17:13 newfile
drwxrwxrwx  2 root   root   4096 Jan 13 13:20 qtree1
drwxrwxrwx  2 root   root   4096 Jan 13 13:13 qtree2
drwxr-xr-x  2 nfs4   daemon 4096 Jan 11 14:30 testdir
```

The `nfsidmap -l` output shows the user `pcuser` in the display but not `netapp-user`; this is the anonymous user in our export-policy rule (`65534`).

```
# nfsidmap -l
6 .id_resolver keys found:
  gid:pcuser@CVSDEMO.LOCAL
  uid:pcuser@CVSDEMO.LOCAL
  gid:daemon@CVSDEMO.LOCAL
  uid:nfs4@CVSDEMO.LOCAL
  gid:root@CVSDEMO.LOCAL
  uid:root@CVSDEMO.LOCAL
```

Next: SMB.

SMB

Previous: NFS.

SMB is a network file sharing protocol developed by Microsoft that provides centralized user/group authentication, permissions, locking, and file sharing to multiple SMB clients over an Ethernet network. Files and folders are presented to clients by way of shares, which can be configured with a variety of share properties and offers access control through share-level permissions. SMB can be presented to any client that offers support for the protocol, including Windows, Apple, and Linux clients.

Cloud Volumes Service provides support for the SMB 2.1 and 3.x versions of the protocol.

Access control/SMB shares

- When a Windows username requests access to the Cloud Volumes Service volume, Cloud Volumes Service looks for a UNIX username using the methods configured by Cloud Volumes Service

administrators.

- If an external UNIX identity provider (LDAP) is configured and Windows/UNIX usernames are identical, then Windows usernames will map 1:1 to UNIX usernames without any additional configuration needed. When LDAP is enabled, Active Directory is used to host those UNIX attributes for user and group objects.
- If Windows names and UNIX names do not match identically, then LDAP must be configured to allow Cloud Volumes Service to use the LDAP name mapping configuration (see the section "Using LDAP for asymmetric name mapping").
- If LDAP is not in use, then Windows SMB users map to a default local UNIX user named `pcuser` in Cloud Volumes Service. This means files written in Windows by users that map to the `pcuser` show UNIX ownership as `pcuser` in multiprotocol NAS environments. `pcuser` here is effectively the `nobody` user in Linux environments (UID 65534).

In deployments with SMB only, the `pcuser` mapping still occurs, but it won't matter, because Windows user and group ownership is correctly displayed and NFS access to the SMB-only volume is not allowed. In addition, SMB-only volumes do not support conversion to NFS or dual-protocol volumes after they are created.

Windows leverages Kerberos for username authentication with the Active Directory domain controllers, which requires a username/password exchange with the AD DCs, which is external to the Cloud Volumes Service instance. Kerberos authentication is used when the `\\SERVERNAME` UNC path is used by the SMB clients and the following is true:

- DNS A/AAAA entry exists for SERVERNAME
- A valid SPN for SMB/CIFS access exists for SERVERNAME

When a Cloud Volumes Service SMB volume is created, the machine account name is created as defined in the section "How Cloud Volumes Service shows up in Active Directory." That machine account name also becomes the SMB share access path because Cloud Volumes Service leverages Dynamic DNS (DDNS) to create the necessary A/AAAA and PTR entries in DNS and the necessary SPN entries on the machine account principal.

For PTR entries to be created, the reverse lookup zone for the Cloud Volumes Service instance IP address must exist on the DNS server.

For example, this Cloud Volumes Service volume uses the following UNC share path: `\\cvs-east-433d.cvsdemo.local`.

In Active Directory, these are the Cloud Volumes Service-generated SPN entries:

```
PS C:\> setspn /L CVS-EAST-433D
Registered ServicePrincipalNames for CN=CVS-EAST-433D,CN=Computers,DC=cvsdemo,DC=local:
        HOST/cvs-east-433d.cvsdemo.local
        HOST/CVS-EAST-433D
```

This is the DNS forward/reverse lookup result:

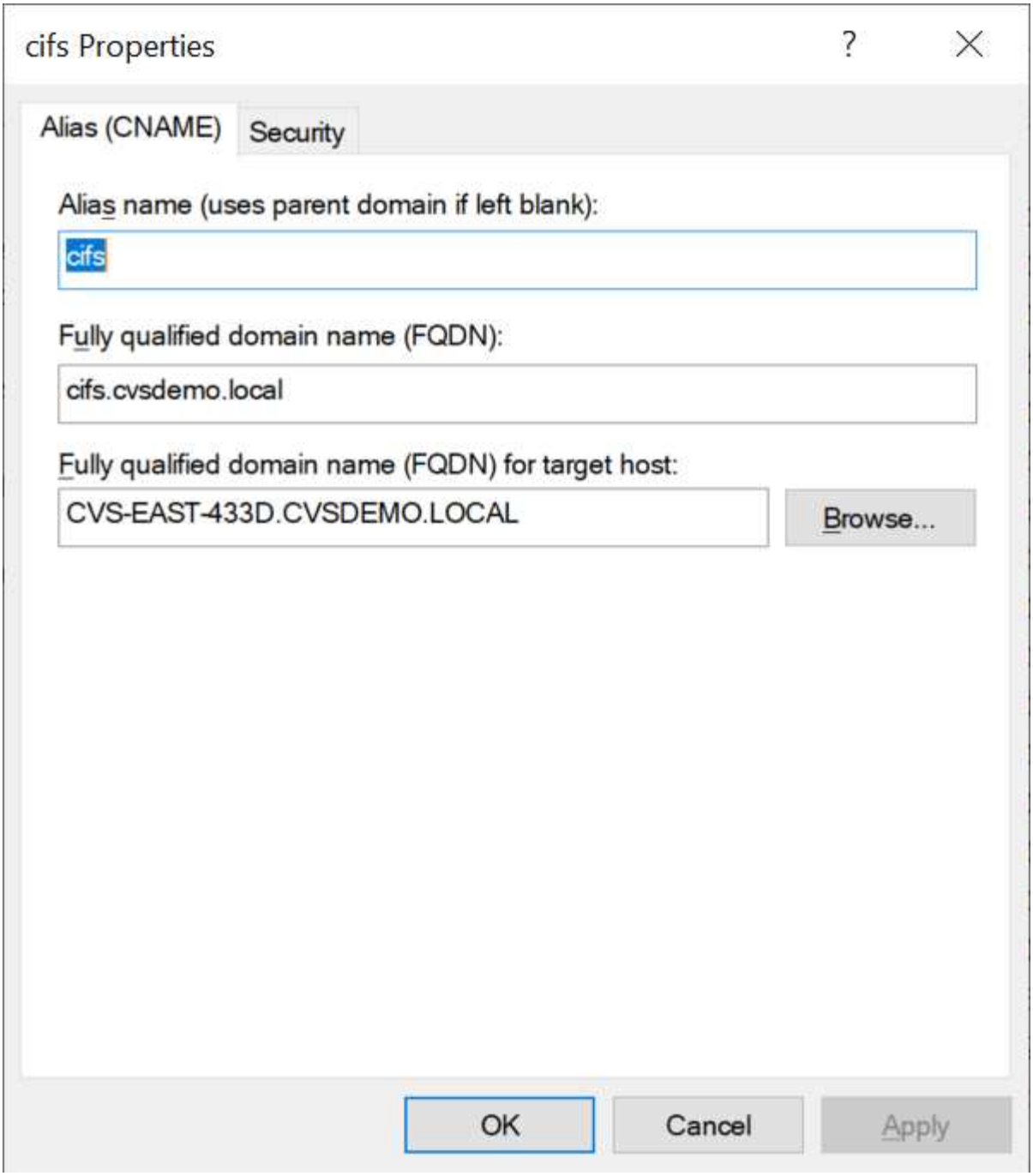
```
PS C:\> nslookup CVS-EAST-433D
Server:  activedirectory. region. lab. internal
Address:  10. xx.0. xx
Name:    CVS-EAST-433D.cvsdemo.local
Address:  10. xxx.0. x
PS C:\> nslookup 10. xxx.0. x
Server:  activedirectory.region.lab.internal
Address:  10.xx.0.xx
Name:    CVS-EAST-433D.CVSDEMO.LOCAL
Address:  10. xxx.0. x
```

Optionally, more access control can be applied by enabling/requiring SMB encryption for SMB shares in Cloud Volumes Service. If SMB encryption isn't supported by one of the endpoints, then access is not allowed.

Using SMB name aliases

In some cases, it might be a security concern for end users to know the machine account name in use for Cloud Volumes Service. In other cases, you might simply want to provide a simpler access path to your end users. In those cases, you can create SMB aliases.

If you want to create aliases for the SMB share path, you can leverage what is known as a CNAME record in DNS. For example, if you want to use the name `\\CIFS` to access shares instead of `\\cvs-east-433d.cvsdemo.local`, but you still want to use Kerberos authentication, a CNAME in DNS that points to the existing A/AAAA record and an additional SPN added to the existing machine account provides Kerberos access.



This is the resulting DNS forward lookup result after adding a CNAME:

```
PS C:\> nslookup cifs
Server:  ok-activedirectory.us-east4-a.c.cv-solution-architect-
lab.internal
Address:  10. xx.0. xx
Name:    CVS-EAST-433D.cvsdemo.local
Address:  10. xxx.0. x
Aliases:  cifs.cvsdemo.local
```

This is the resulting SPN query after adding new SPNs:

```
PS C:\> setspn /L CVS-EAST-433D
Registered ServicePrincipalNames for CN=CVS-EAST-433D,CN=Computers,DC=cvsdemo,DC=local:
        cifs/cifs.cvsdemo.local
        cifs/cifs
        HOST/cvs-east-433d.cvsdemo.local
        HOST/CVS-EAST-433D
```

In a packet capture, we can see the Session Setup Request using the SPN tied to the CNAME.

```
431 4.156722                                    SMB2     308 Negotiate Protocol Response
432 4.156785                                    SMB2     232 Negotiate Protocol Request
434 4.158108                                    SMB2     374 Negotiate Protocol Response
435 4.160977                                    SMB2    1978 Session Setup Request
437 4.166224                                    SMB2     322 Session Setup Response
438 4.166891                                    SMB2     152 Tree Connect Request Tree: \\cifs\IPC$
439 4.168063                                    SMB2     138 Tree Connect Response
```

```
realm: CVSDEMO.LOCAL
sname
   name-type: kRB5-NT-SRV-INST (2)
   sname-string: 2 items
      SNameString: cifs
      SNameString: cifs
enc-part
   etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

SMB authentication dialects

Cloud Volumes Service supports the following dialects for SMB authentication:

- LM
- NTLM
- NTLMv2
- Kerberos

Kerberos authentication for SMB share access is the most secure level of authentication you can use. With AES and SMB encryption enabled, the security level is further increased.

Cloud Volumes Service also supports backward compatibility for LM and NTLM authentication. When Kerberos is misconfigured (such as when creating SMB aliases), share access falls back to weaker authentication methods (such as NTLMv2). Because these mechanisms are less secure, they are disabled in some Active Directory environments. If weaker authentication methods are disabled and Kerberos is not configured properly, share access fails because there is no valid authentication method to fall back to.

For information about configuring/viewing your supported authentication levels in Active Directory, see Network security: LAN Manager authentication level.

Permission models

NTFS/File permissions

NTFS permissions are the permissions applied to files and folders in file systems adhering to NTFS logic. You can apply NTFS permissions in `Basic` or `Advanced` and can be set to `Allow` or `Deny` for access control.

Basic permissions include the following:

- Full Control
- Modify
- Read & Execute
- Read
- Write

When you set permissions for a user or group, referred to as an ACE, it resides in an ACL. NTFS permissions use the same read/write/execute basics as UNIX mode bits, but they can also extend to more granular and extended access controls (also known as Special Permissions), such as Take Ownership, Create Folders/Append Data, Write Attributes, and more.

Standard UNIX mode bits do not provide the same level of granularity as NTFS permissions (such as being able to set permissions for individual user and group objects in an ACL or setting extended attributes). However, NFSv4.1 ACLs do provide the same functionality as NTFS ACLs.

NTFS permissions are more specific than share permissions and can be used in conjunction with share permissions. With NTFS permission structures, the most restrictive applies. As such, explicit denials to a user or group overrides even Full Control when defining access rights.

NTFS permissions are controlled from Windows SMB clients.

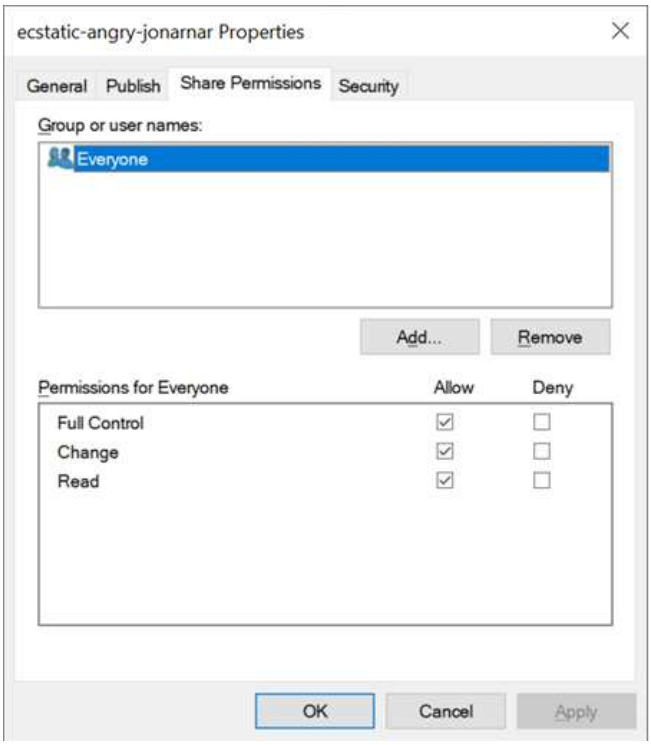
Share permissions

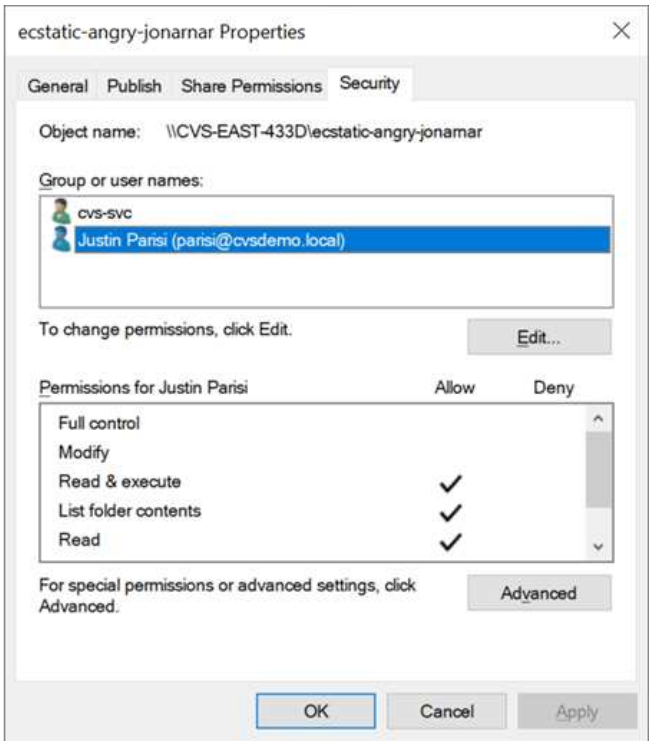
Share permissions are more general than NTFS permissions (Read/Change/Full Control only) and control the initial entry into an SMB share—similar to how NFS export policy rules work.

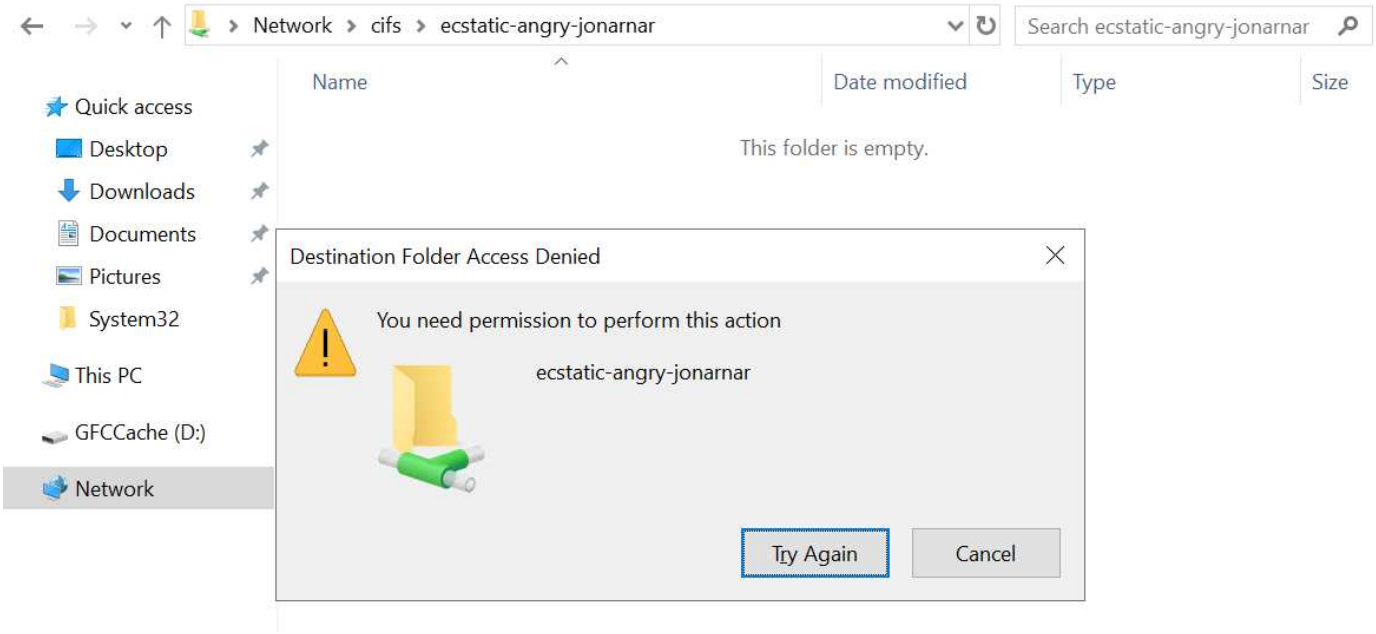
Although NFS export policy rules control access through host-based information such as IP addresses or host names, SMB share permissions can control access by using user and group ACEs in a share ACL. You can set share ACLs either from the Windows client or from the Cloud Volumes Service management UI.

By default, share ACLs and initial volume ACLs include Everyone with Full Control. The file ACLs should be changed but share permissions are overruled by the file permissions on objects in the share.

For instance, if a user is only allowed Read access to the Cloud Volumes Service volume file ACL, they are denied access to create files and folders even though the share ACL is set to Everyone with Full Control, as shown in the following figure.







For best security results, do the following:

- Remove Everyone from the share and file ACLs and instead set share access for users or groups.
- Use groups for access control instead of individual users for ease of management and faster removal/addition of users to share ACLs through group management.
- Allow less restrictive, more general share access to the ACEs on the share permissions and lock down access to users and groups with file permissions for more granular access control.
- Avoid general use of explicit deny ACLs, because they override allow ACLs. Limit use of explicit deny ACLs for users or groups that need to be restricted from access to a file system quickly.
- Make sure that you pay attention to the ACL inheritance settings when modifying permissions; setting the inheritance flag at the top level of a directory or volume with high file counts means that each file below that

directory or volume has inherited permissions added to it, which can create unwanted behavior such as unintended access/denial and long churn of permission modification as each file is adjusted.

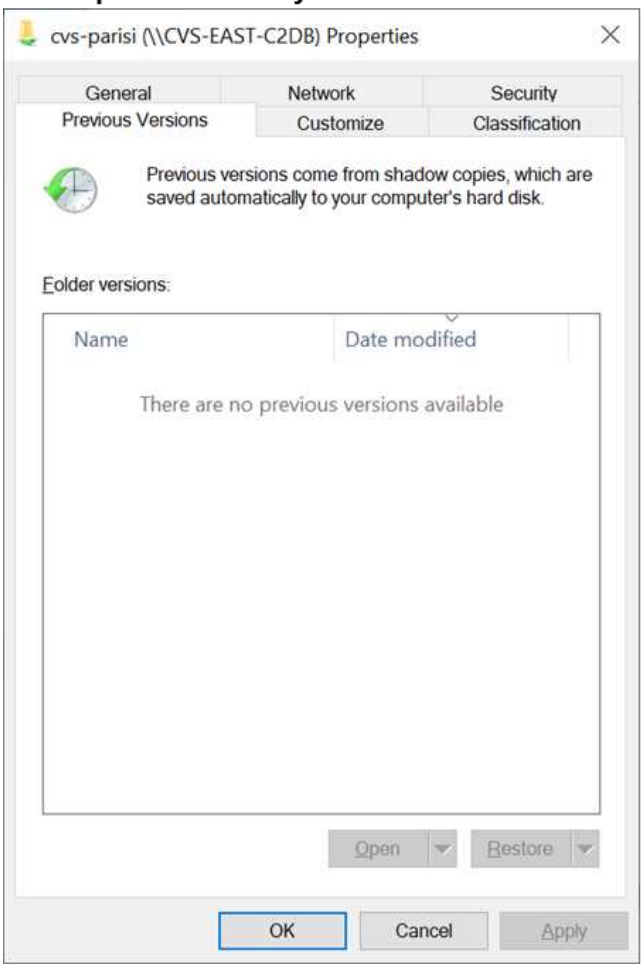
SMB share security features

When you first create a volume with SMB access in Cloud Volumes Service, you are presented with a series of choices for securing that volume.

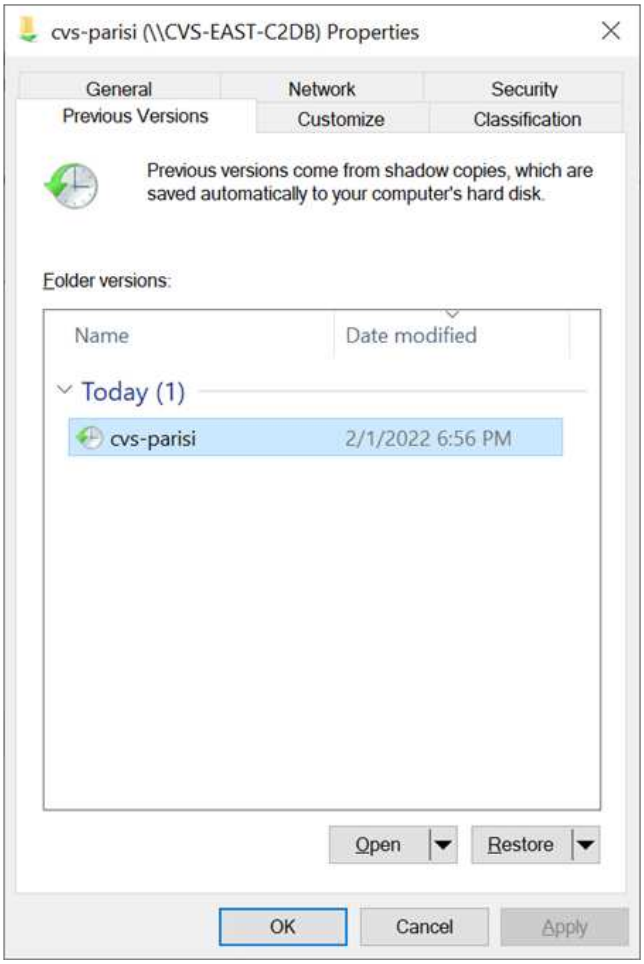
Some of these choices depend on the Cloud Volumes Service level (Performance or Software) and choices include:

- **Make snapshot directory visible (available for both CVS-Performance and CVS-SW).** This option controls whether or not SMB clients can access the Snapshot directory in an SMB share (`\\server\share\~snapshot` and/or Previous Versions tab). The default setting is Not Checked, which means that the volume defaults to hiding and disallowing access to the `~snapshot` directory, and no Snapshot copies appear in the Previous Versions tab for the volume.

Snapshot directory visible: Unchecked



Snapshot directory visible: Checked



Hiding Snapshot copies from end users might be desired for security reasons, performance reasons (hiding these folders from AV scans) or preference. Cloud Volumes Service Snapshots are read- only, so even if these Snapshots are visible, end users cannot delete or modify files in the Snapshot directory. File permissions on the files or folders at the time the Snapshot copy was taken apply. If a file or folder's permissions change between Snapshot copies, then the changes also apply to the files or folders in the Snapshot directory. Users and groups can gain access to these files or folders based on permissions. While deletes or modifications of files in the Snapshot directory are not possible, it is possible to copy files or folders out of the Snapshot

directory.

- **Enable SMB encryption (available for both CVS-Performance and CVS-SW).** SMB encryption is disabled on the SMB share by default (unchecked). Checking the box enables SMB encryption, which means traffic between the SMB client and server is encrypted in-flight with the highest supported encryption levels negotiated. Cloud Volumes Service supports up to AES-256 encryption for SMB. Enabling SMB encryption does carry a performance penalty that might or might not be noticeable to your SMB clients—roughly in the 10-20% range. NetApp strongly encourages testing to see if that performance penalty is acceptable.
- **Hide SMB share (available for both CVS-Performance and CVS-SW).** Setting this option hides the SMB share path from normal browsing. This means that clients that do not know the share path cannot see the shares when accessing the default UNC path (such as `\\CVS-SMB`). When the checkbox is selected, only clients that explicitly know the SMB share path or have the share path defined by a Group Policy Object can access it (security through obfuscation).
- **Enable access-based enumeration (ABE) (CVS-SW only).** This is similar to hiding the SMB share, except the shares or files are only hidden from users or groups that do not have permissions to access the objects. For instance, if Windows user `joe` is not allowed at least Read access through the permissions, then the Windows user `joe` cannot see the SMB share or files at all. This is disabled by default, and you can enable it by selecting the checkbox. For more information on ABE, see the NetApp Knowledge Base article How does Access Based Enumeration (ABE) work?
- **Enable Continuously Available (CA) share support (CVS-Performance only).** Continuously Available SMB shares provide a way to minimize application disruptions during failover events by replicating lock states across nodes in the Cloud Volumes Service backend system. This is not a security feature, but it does offer better overall resiliency. Currently, only SQL Server and FSLogix applications are supported for this functionality.

Default hidden shares

When an SMB server is created in Cloud Volumes Service, there are hidden administrative shares (using the $ naming convention) that are created in addition to the data volume SMB share. These include C$ (namespace access) and IPC$ (sharing named pipes for communication between programs, such as the remote procedure calls (RPC) used for Microsoft Management Console (MMC) access).

The IPC$ share contains no share ACLs and cannot be modified—it is strictly used for RPC calls and Windows disallows anonymous access to these shares by default.

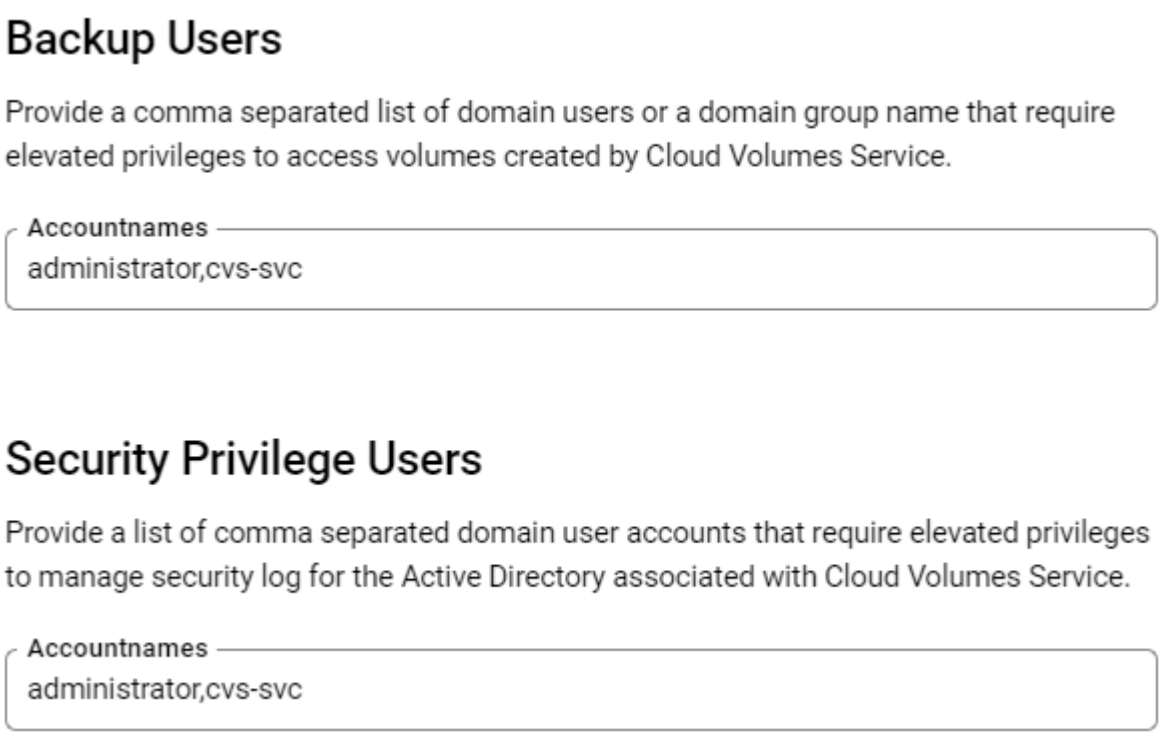
The C$ share allows BUILTIN/Administrators access by default, but Cloud Volumes Service automation removes the share ACL and does not allow access to anyone because access to the C$ share allows visibility into all mounted volumes in the Cloud Volumes Service file systems. As a result, attempts to navigate to `\\SERVER\C$` fail.

Accounts with local/BUILTIN administrator/backup rights

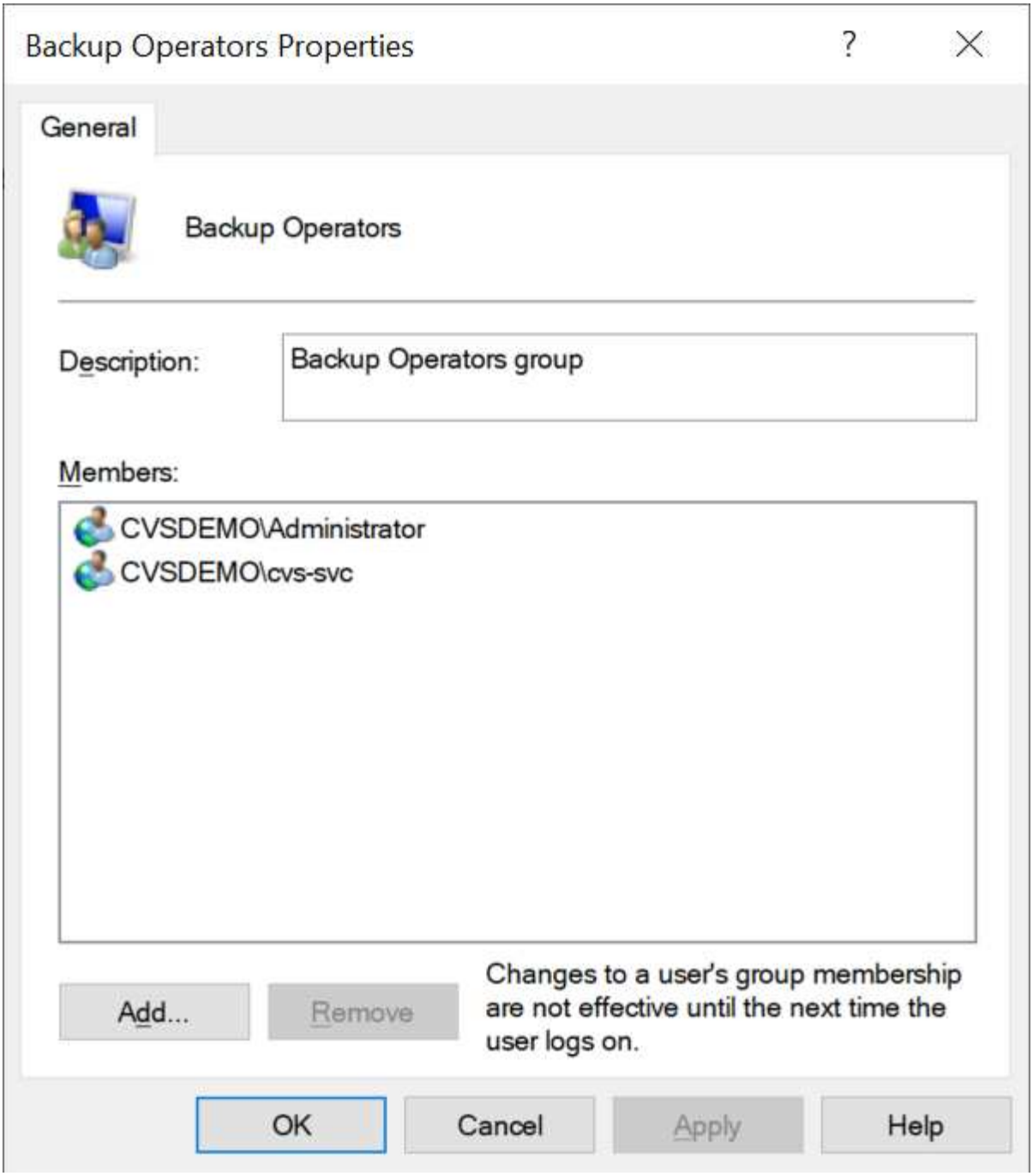
Cloud Volumes Service SMB servers maintain similar functionality to regular Windows SMB servers in that there are local groups (such as BUILTIN\Administrators) that apply access rights to select domain users and groups.

When you specify a user to be added to Backup Users, the user is added to the BUILTIN\Backup Operators group in the Cloud Volumes Service instance that uses that Active Directory connection, which then gets the SeBackupPrivilege and SeRestorePrivilege.

When you add a user to Security Privilege Users, the user is given the SeSecurityPrivilege, which is useful in some application use cases, such as SQL Server on SMB shares.



You can view Cloud Volumes Service local group memberships through the MMC with the proper privileges. The following figure shows users that have been added by using the Cloud Volumes Service console.

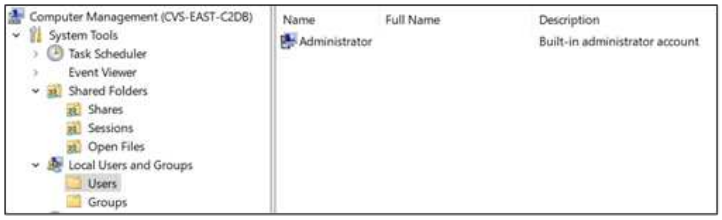


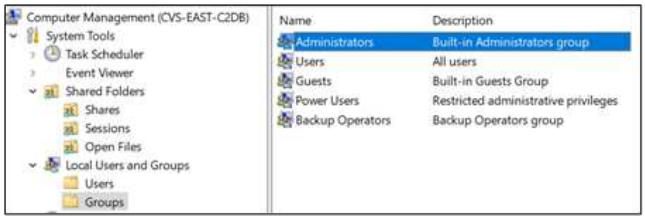
The following table shows the list of default BUILTIN groups and what users/groups are added by default.

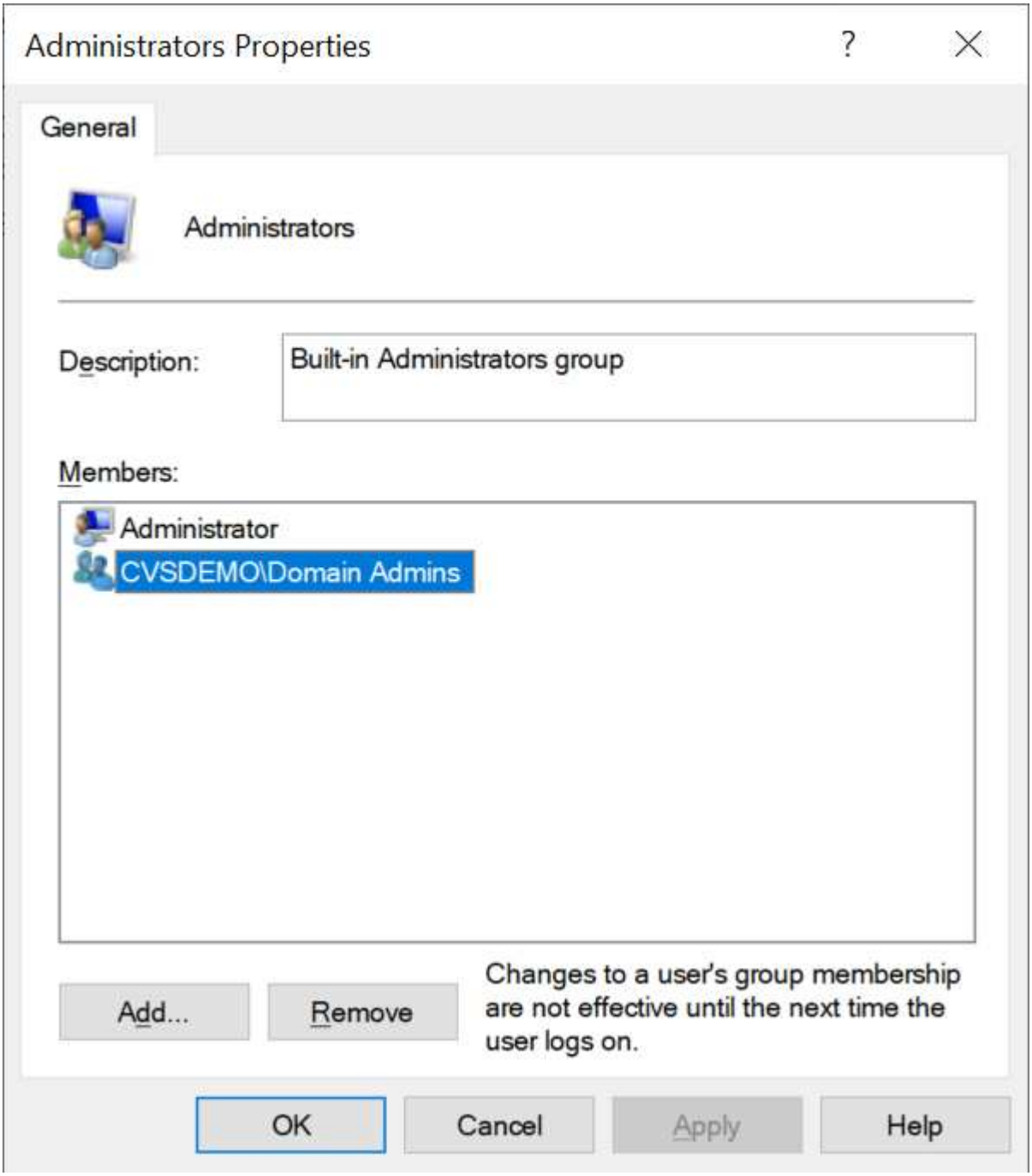
Local/BUILTIN group	Default members
BUILTIN\Administrators*	DOMAIN\Domain Admins
BUILTIN\Backup Operators*	None
BUILTIN\Guests	DOMAIN\Domain Guests
BUILTIN\Power Users	None
BUILTIN\Domain Users	DOMAIN\Domain Users

*Group membership controlled in Cloud Volumes Service Active Directory connection configuration.

You can view local users and groups (and group members) in the MMC window, but you cannot add or delete objects or change group memberships from this console. By default, only the Domain Admins group and Administrator are added to the BUILTIN\Administrators group in Cloud Volumes Service. Currently, you cannot modify this.







MMC/Computer Management access

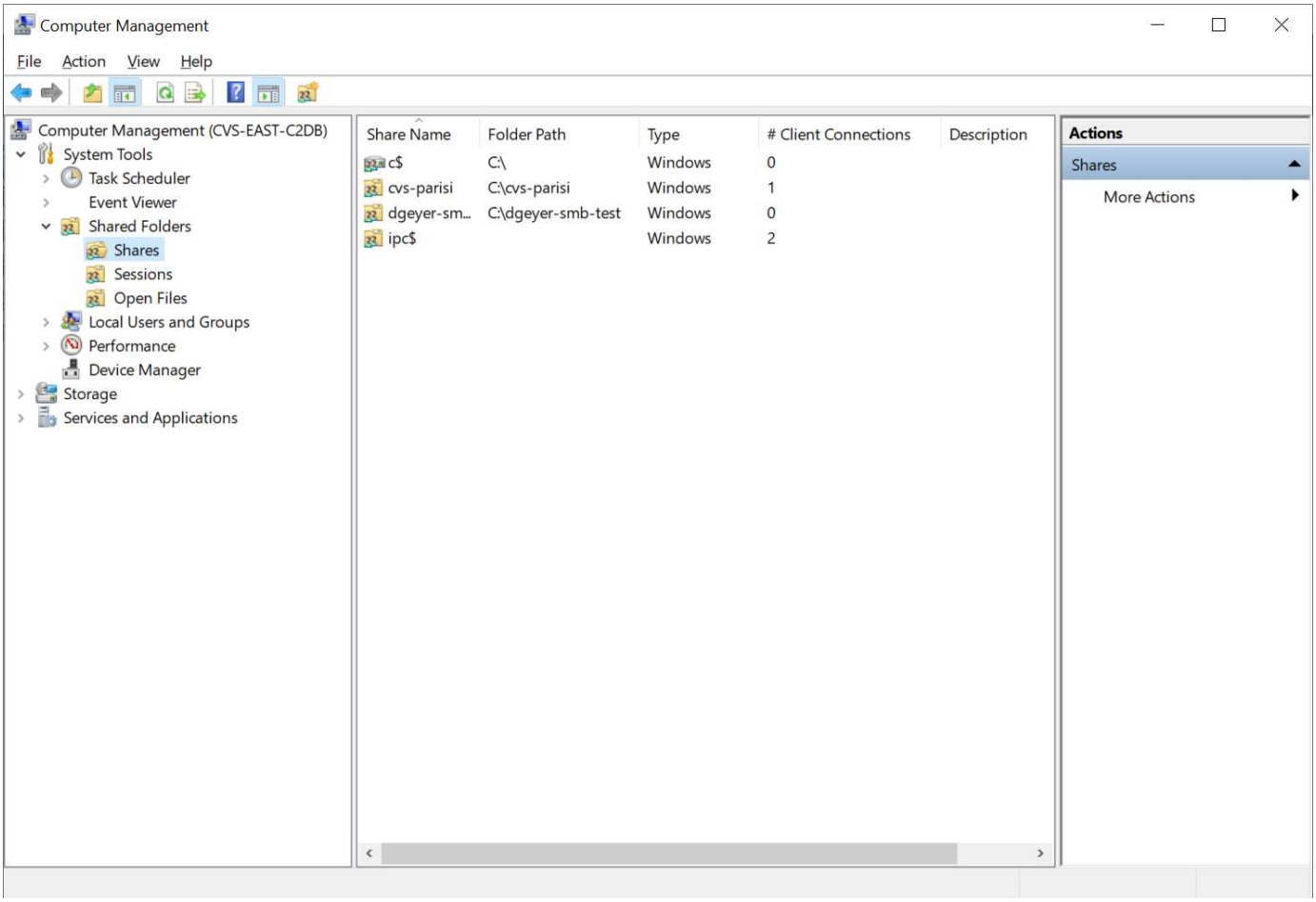
SMB access in Cloud Volumes Service provides connectivity to the Computer Management MMC, which allows you to view shares, manage share ACLs, ands view/manage SMB sessions and open files.

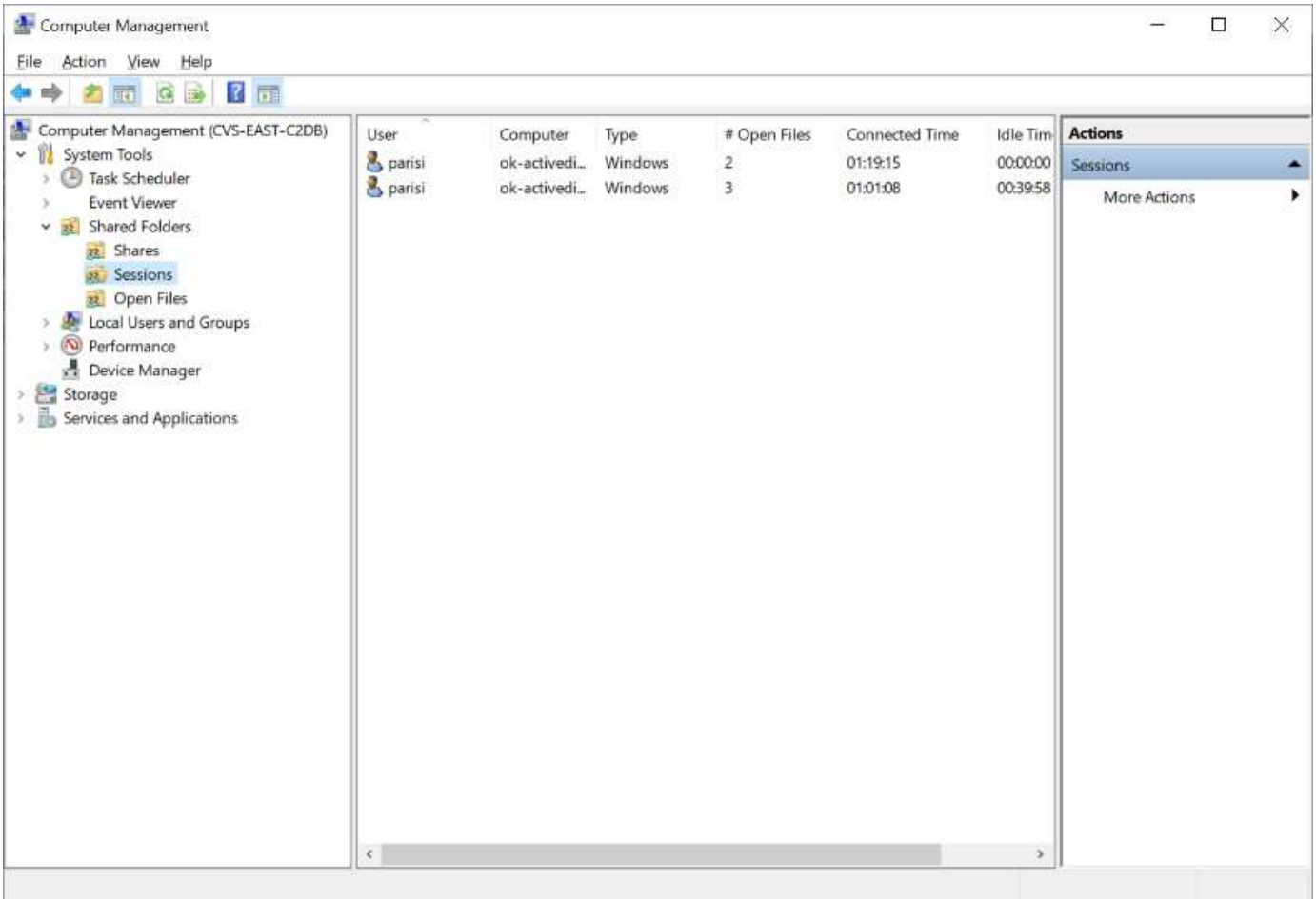
To use the MMC to view SMB shares and sessions in Cloud Volumes Service, the user logged in currently must be a domain administrator. Other users are allowed access to view or manage the SMB server from MMC and receive a You Do Not Have Permissions dialog box when attempting to view shares or sessions on the Cloud Volumes Service SMB instance.

To connect to the SMB server, open Computer Management, right click Computer Management and then select Connect To Another Computer. This opens the Select Computer dialog box where you can enter the SMB server name (found in the Cloud Volumes Service volume information).

When you view SMB shares with the proper permissions, you see all available shares in the Cloud Volumes Service instance that share the Active Directory connection. To control this behavior, set the Hide SMB Shares option on the Cloud Volumes Service volume instance.

Remember, only one Active Directory connection is allowed per region.





The following table shows a list of supported/unsupported functionality for the MMC.

Supported functions	Unsupported functions
• View shares • View active SMB sessions • View open files • View local users and groups • View local group memberships • Enumerate the list of sessions, files, and tree connections in the system • Close open files in the system • Close open sessions • Create/manage shares	• Creating new local users/groups • Managing/viewing existing local user/groups • View events or performance logs • Managing storage • Managing services and applications

SMB server security information

The SMB server in Cloud Volumes Service uses a series of options that define security policies for SMB connections, including things such as Kerberos clock skew, ticket age, encryption, and more.

The following table contains a list of those options, what they do, the default configurations, and if they can be

modified with Cloud Volumes Service. Some options do not apply to Cloud Volumes Service.

Security option	What it does	Default value	Can change?
Maximum Kerberos Clock Skew (minutes)	Maximum time skew between Cloud Volumes Service and domain controllers. If the time skew exceeds 5 minutes, Kerberos authentication fails. This is set to the Active Directory default value.	5	No
Kerberos Ticket Lifetime (hours)	Maximum time a Kerberos ticket remains valid before requiring a renewal. If no renewal occurs before the 10 hours, you must obtain a new ticket. Cloud Volumes Service performs these renewals automatically. 10 hours is the Active Directory default value.	10	No
Maximum Kerberos Ticket Renewal (days)	Maximum number of days that a Kerberos ticket can be renewed before a new authorization request is needed. Cloud Volumes Service automatically renews tickets for SMB connections. Seven days is the Active Directory default value.	7	No
Kerberos KDC Connection Timeout (secs)	The number of seconds before a KDC connection times out.	3	No
Require Signing for Incoming SMB Traffic	Setting to require signing for SMB traffic. If set to true, clients that do not support signing fail connectivity.	False	
Require Password Complexity for Local User Accounts	Used for passwords on local SMB users. Cloud Volumes Service does not support local user creation, so this option does not apply to Cloud Volumes Service.	True	No

Security option	What it does	Default value	Can change?
Use start_tls for Active Directory LDAP Connections	Used to enable start TLS connections for Active Directory LDAP. Cloud Volumes Service does not currently support enabling this.	False	No
Is AES-128 and AES-256 Encryption for Kerberos Enabled	This controls whether AES encryption is used for Active Directory connections and is controlled with the Enable AES Encryption for Active Directory Authentication option when creating/modifying the Active Directory connection.	False	Yes
LM Compatibility Level	Level of supported authentication dialects for Active Directory connections. See the section “SMB authentication dialects” for more information.	ntlmv2-krb	No
Require SMB Encryption for Incoming CIFS Traffic	Requires SMB encryption for all shares. This is not used by Cloud Volumes Service; instead, set encryption on a per-volume basis (see the section “SMB share security features”).	False	No
Client Session Security	Sets signing and/or sealing for LDAP communication. This is not currently set in Cloud Volumes Service but might be needed in future releases to address . Remediation for LDAP authentication issues due to the Windows patch is covered in the section “LDAP channel binding.”.	None	No
SMB2 enable for DC connections	Uses SMB2 for DC connections. Enabled by default.	System-default	No

Security option	What it does	Default value	Can change?
LDAP Referral Chasing	When using multiple LDAP servers, referral chasing allows the client to refer to other LDAP servers in the list when an entry is not found in the first server. This is currently not supported by Cloud Volumes Service.	False	No
Use LDAPS for Secure Active Directory Connections	Enables the use of LDAP over SSL. Currently not supported by Cloud Volumes Service.	False	No
Encryption is required for DC Connection	Requires encryption for successful DC connections. Disabled by default in Cloud Volumes Service.	False	No

Next: Dual-protocol/multiprotocol.

Dual-protocol/multiprotocol

Previous: SMB.

Cloud Volumes Service offers the ability to share the same datasets to both SMB and NFS clients while maintaining proper access permissions (dual-protocol). This is done by coordinating identity mapping between protocols and using a centralized backend LDAP server to provide the UNIX identities to Cloud Volumes Service. You can use Windows Active Directory to provide both Windows and UNIX users for ease of use.

Access control

- **Share access controls.** Determine which clients and/or user and groups can access a NAS share. For NFS, export policies and rules control client access to exports. NFS exports are managed from the Cloud Volumes Service instance. SMB makes use of CIFS/SMB shares and share ACLs to provide more granular control at the user and group level. You can only configure share-level ACLs from SMB clients by using MMC/Computer Management with an account that has administrator rights on the Cloud Volumes Service instance (see the section “Accounts with local/BUILTIN administrator/backup rights.”).
- **File access controls.** Control permissions at a file or folder level and are always managed from the NAS client. NFS clients can make use of traditional mode bits (rwx) or NFSv4 ACLs. SMB clients leverage NTFS permissions.

The access control for volumes that serve data to both NFS and SMB depends on the protocol in use. For information on permissions with dual protocol, see the section “Permission model.”

User mapping

When a client accesses a volume, Cloud Volumes Service attempts to map the incoming user to a valid user in the opposite direction. This is necessary for proper access to be determined across protocols and to ensure that the user requesting access is indeed who they claim to be.

For example, if a Windows user named `joe` attempts access to a volume with UNIX permissions through SMB, then Cloud Volumes Service performs a search to find a corresponding UNIX user named `joe`. If one exists, then files that are written to an SMB share as Windows user `joe` appears as UNIX user `joe` from NFS clients.

Alternately, if a UNIX user named `joe` attempts access to a Cloud Volumes Service volume with Windows permissions, then the UNIX user must be able to map to a valid Windows user. Otherwise, access to the volume is denied.

Currently, only Active Directory is supported for external UNIX identity management with LDAP. For more information about configuring access to this service, see Creating an AD connection.

Permission model

When using dual-protocol setups, Cloud Volumes Service makes use of security styles for volumes to determine the type of ACL. These security styles are set based on which NAS protocol is specified, or in the case of dual protocol, is a choice made at the time of Cloud Volumes Service volume creation.

- If you are only using NFS, Cloud Volumes Service volumes use UNIX permissions.
- If you are only using SMB, Cloud Volumes Service volumes use NTFS permissions.

If you are creating a dual-protocol volume, you can choose the ACL style at volume creation. This decision should be made based on the desired permissions management. If your users manage permissions from Windows/SMB clients, select NTFS. If your users prefer using NFS clients and chmod/chown, use UNIX security styles.

Next: Considerations for creating Active Directory connections.

Considerations for creating Active Directory connections

Previous: Dual-protocol/multiprotocol.

Cloud Volumes Service provides the ability to connect your Cloud Volumes Service instance to an external Active Directory server for identity management for both SMB and UNIX users. Creating an Active Directory connection is required to use SMB in Cloud Volumes Service.

The configuration for this provides several options that require some consideration for security. The external Active Directory server can be an on-premises instance or cloud native. If you are using an on-premises Active Directory server, don’t expose the domain to the external network (such as with a DMZ or an external IP address). Instead, use secure private tunnels or VPNs, one-way forest trusts, or dedicated network connections to the on-premises networks with Private Google Access. See the Google Cloud documentation for more information about best practices using Active Directory in Google Cloud.

CVS-SW requires Active Directory servers to be located in the same region. If a DC connection is attempted in CVS-SW to another region, the attempt fails. When using CVS-SW, be sure to create Active Directory sites that include the Active Directory DCs and then specify sites in Cloud Volumes Service to avoid cross-region DC connection attempts.

Active Directory credentials

When SMB or LDAP for NFS is enabled, Cloud Volumes Service interacts with the Active Directory controllers to create a machine account object to use for authentication. This is no different from how a Windows SMB client joins a domain and requires the same access rights to Organizational Units (OUs) in Active Directory.

In many cases, security groups do not allow the use of a Windows administrator account on external servers such as Cloud Volumes Service. In some cases, the Windows Administrator user is disabled entirely as a security best practice.

Permissions needed to create SMB machine accounts

To add Cloud Volumes Service machine objects to an Active Directory, an account that either has administrative rights to the domain or has delegated permissions to create and modify machine account objects to a specified OU is required. You can do this with the Delegation of Control Wizard in Active Directory by creating a custom task that provides a user access to creation/deletion of computer objects with the following access permissions provided:

- Read/Write
- Create/Delete All Child Objects
- Read/Write All Properties
- Change/Reset Password

Doing this automatically adds a security ACL for the defined user to the OU in Active Directory and minimizes the access to the Active Directory environment. After a user has been delegated, that username and password can be provided as Active Directory Credentials in this window.

The username and password that is passed to the Active Directory domain leverages Kerberos encryption during the machine account object query and creation for added security.

Active Directory connection details

The Active Directory Connection Details provide fields for administrators to give specific Active Directory schema information for machine account placement, such as the following:

- **Active Directory Connection Type.** Used to specify whether the Active Directory connection in a region is used for volumes of either Cloud Volumes Service or CVS-Performance service type. If this is set incorrectly on an existing connection, it might not work properly when used or edited.
- **Domain.** The Active Directory domain name.
- **Site.** Limits Active Directory servers to a specific site for security and performance considerations. This is necessary when multiple Active Directory servers span regions because Cloud Volumes Service does not currently support allowing Active Directory authentication requests to Active Directory servers in a different region than the Cloud Volumes Service instance. (For instance, the Active Directory domain controller is in a region that only CVS-Performance supports but you want an SMB share in a CVS-SW instance.)
- **DNS servers.** DNS servers to use in name lookups.

- **NetBIOS name (optional).** If desired, the NetBIOS name for the server. This what is used when new machine accounts are created using the Active Directory connection. For instance, if the NetBIOS name is set to CVS-EAST then the machine account names will be CVS-EAST-{1234}. See the section "How Cloud Volumes Service shows up in Active Directory" for more information.
- **Organizational Unit (OU).** The specific OU to create the computer account. This is useful if you're delegating control to a user for machine accounts to a specific OU.
- **AES Encryption.** You can also check or uncheck the Enable AES Encryption for AD Authentication checkbox. Enabling AES encryption for Active Directory authentication provides extra security for Cloud Volumes Service to Active Directory communication during user and group lookups. Before enabling this option, check with your domain administrator to confirm that the Active Directory domain controllers support AES authentication.

By default, most Windows servers do not disable weaker ciphers (such as DES or RC4-HMAC), but if you choose to disable weaker ciphers, confirm Cloud Volumes Service Active Directory connection has been configured to enable AES. Otherwise, authentication failures occur. Enabling AES encryption doesn't disable weaker ciphers but instead adds support for AES ciphers to the Cloud Volumes Service SMB machine account.

Kerberos realm details

This option does not apply to SMB servers. Rather, it is used when configuring NFS Kerberos for the Cloud Volumes Service system. When these details are populated, the NFS Kerberos realm is configured (similar to a krb5.conf file on Linux) and is used when NFS Kerberos is specified on the Cloud Volumes Service volume creation, as the Active Directory connection acts as the NFS Kerberos Distribution Center (KDC).

Non-Windows KDCs are currently unsupported for use with Cloud Volumes Service.

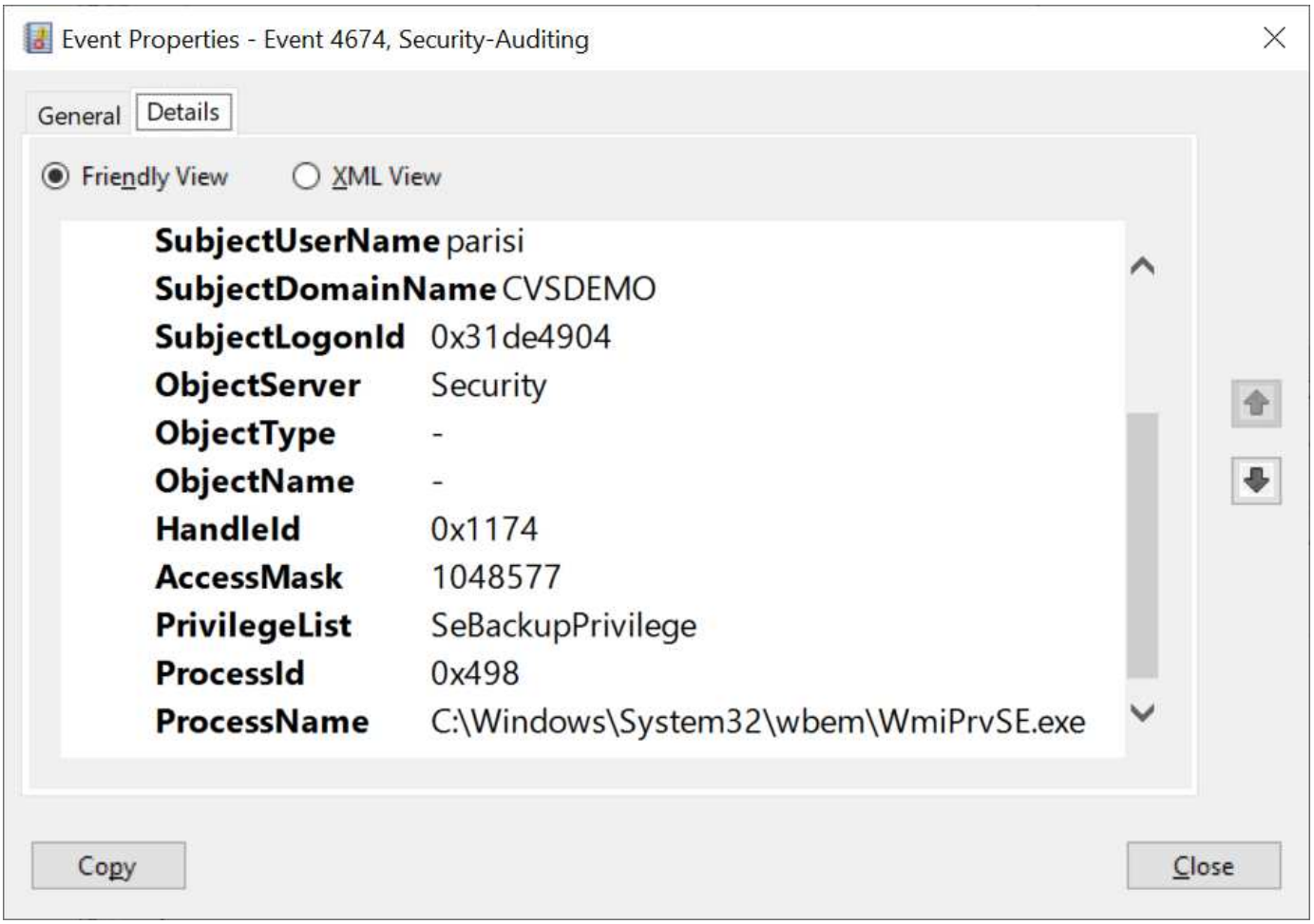
Region

A region enables you to specify the location where the Active Directory connection resides. This region must be the same region as the Cloud Volumes Service volume.

- **Local NFS Users with LDAP.** In this section, there is also an option to Allow Local NFS Users with LDAP. This option must be left unselected if you want to extend your UNIX user group membership support beyond the 16-group limitation of NFS (extended groups). However, using extended groups requires a configured LDAP server for UNIX identities. If you don't have an LDAP server, leave this option unselected. If you have an LDAP server and want to also use local UNIX users (such as root), select this option.

Backup users

This option enables you to specify Windows users that have backup permissions to the Cloud Volumes Service volume. Backup privileges (SeBackupPrivilege) are necessary for some applications to properly backup and restore data in NAS volumes. This user has a high level of access to data in the volume, so you should consider enabling auditing of that user access. After it is enabled, audit events display in Event Viewer > Windows Logs > Security.



Security privilege users

This option enables you to specify Windows users that have security modification permissions to the Cloud Volumes Service volume. Security privileges (SeSecurityPrivilege) are necessary for some applications (such as SQL Server) to properly set permissions during installation. This privilege is needed to manage the security log. Although this privilege is not as powerful as SeBackupPrivilege, NetApp recommends auditing user access of users with this privilege level if needed.

For more information, see Special privileges assigned to new logon.

How Cloud Volumes Service shows up in Active Directory

Cloud Volumes Service shows up in Active Directory as a normal machine account object. The naming conventions are as follows.

- CIFS/SMB and NFS Kerberos create separate machine account objects.
- NFS with LDAP enabled creates a machine account in Active Directory for Kerberos LDAP binds.
- Dual protocol volumes with LDAP share the CIFS/SMB machine account for LDAP and SMB.
- CIFS/SMB machine accounts use a naming convention of NAME-1234 (random four digit ID with hyphen appended to <10 character name) for the machine account. You can define NAME by the NetBIOS name setting on the Active Directory connection (see the section “Active Directory connection details”).
- NFS Kerberos uses NFS-NAME-1234 as the naming convention (up to 15 characters). If more than 15 characters are used, the name is NFS-TRUNCATED-NAME-1234.

- NFS-only CVS-Performance instances with LDAP enabled create an SMB machine account for binding to the LDAP server with the same naming convention as CIFS/SMB instances.
- When an SMB machine account is created, default hidden admin shares (see the section "Default hidden shares") are also created (c$, admin$, ipc$), but those shares have no ACLs assigned and are inaccessible.
- The machine account objects are placed in CN=Computers by default, but a you can specify a different OU when necessary. See the section "Permissions needed to create SMB machine accounts" for information about what access rights are needed to add/remove machine account objects for Cloud Volumes Service.

When Cloud Volumes Service adds the SMB machine account to Active Directory, the following fields are populated:

- cn (with the specified SMB server name)
- dNSHostName (with SMBserver.domain.com)
- msDS-SupportedEncryptionTypes (Allows DES_CBC_MD5, RC4_HMAC_MD5 if AES encryption is not enabled; if AES encryption is enabled, DES_CBC_MD5, RC4_HMAC_MD5, AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96 are allowed for Kerberos ticket exchange with the machine account for SMB)
- name (with the SMB server name)
- sAMAccountName (with SMBserver$)
- servicePrincipalName (with host/smbserver.domain.com and host/smbserver SPNs for Kerberos)

If you want to disable weaker Kerberos encryption types (enctype) on the machine account, you can change the msDS-SupportedEncryptionTypes value on the machine account to one of the values in the following table to allow AES only.

msDS-SupportedEncryptionTypes value	Enctype enabled
2	DES_CBC_MD5
4	RC4_HMAC
8	AES128_CTS_HMAC_SHA1_96 only
16	AES256_CTS_HMAC_SHA1_96 only
24	AES128_CTS_HMAC_SHA1_96 and AES256_CTS_HMAC_SHA1_96
30	DES_CBC_MD5, RC4_HMAC, AES128_CTS_HMAC_SHA1_96 and AES256_CTS_HMAC_SHA1_96

To enable AES encryption for SMB machine accounts, click Enable AES Encryption for AD Authentication when creating the Active Directory connection.

To enable AES encryption for NFS Kerberos, see the Cloud Volumes Service documentation.

Next: Other NAS Infrastructure service dependencies (KDC, LDAP, DNS).

Other NAS Infrastructure service dependencies (KDC, LDAP, and DNS)

Previous: Considerations for creating Active Directory connections.

When using Cloud Volumes Service for NAS shares, there might be external dependencies required for proper functionality. These dependencies are in play under specific circumstances. The following table shows various configuration options and what, if any, dependencies are required.

Configuration	Dependencies required
NFSv3 only	None
NFSv3 Kerberos only	Windows Active Directory: * KDC * DNS * LDAP
NFSv4.1 only	Client ID mapping configuration (/etc/idmap.conf)
NFSv4.1 Kerberos only	• Client ID mapping configuration (/etc/idmap.conf) • Windows Active Directory: KDC DNS LDAP
SMB only	Active Directory: * KDC * DNS
Multiprotocol NAS (NFS and SMB)	• Client ID mapping configuration (NFSv4.1 only; /etc/idmap.conf) • Windows Active Directory: KDC DNS LDAP

Kerberos keytab rotation/password resets for machine account objects

With SMB machine accounts, Cloud Volumes Service schedules periodic password resets for the SMB machine account. These password resets occur using Kerberos encryption and operate on a schedule of every fourth Sunday at a random time between 11PM and 1AM. These password resets change the Kerberos key versions, rotate the keytabs stored on the Cloud Volumes Service system, and help maintain a greater level of security for SMB servers running in Cloud Volumes Service. Machine account passwords are randomized and are not known to administrators.

For NFS Kerberos machine accounts, password resets take place only when a new keytab is created/exchanged with the KDC. Currently, this is not possible to do in Cloud Volumes Service.

Network ports for use with LDAP and Kerberos

When using LDAP and Kerberos, you should determine the network ports in use by these services. You can find a complete list of ports in use by Cloud Volumes Service in the Cloud Volumes Service documentation on security considerations.

LDAP

Cloud Volumes Service acts as an LDAP client and uses standard LDAP search queries for user and group lookups for UNIX identities. LDAP is necessary if you intend to use users and groups outside the standard default users provided by Cloud Volumes Service. LDAP is also necessary if you plan on using NFS Kerberos with user principals (such as user1@domain.com). Currently, only LDAP using Microsoft Active Directory is supported.

To use Active Directory as a UNIX LDAP server, you must populate the necessary UNIX attributes on users and groups you intend to use for UNIX identities. Cloud Volumes Service uses a default LDAP schema template that queries attributes based on RFC-2307-bis. As a result, the following table shows the bare minimum necessary Active Directory attributes to populate for users and groups and what each attribute is used for.

For more information about setting LDAP attributes in Active Directory, see Managing dual-protocol access.

Attribute	What it does
uid*	Specifies the UNIX user name
uidNumber*	Specifies the UNIX user's numeric ID
gidNumber*	Specifies the UNIX user's primary group numeric ID
objectClass*	Specifies what type of object is being used; Cloud Volumes Service requires "user" to be included in the list of object classes (is included in most Active Directory deployments by default).
name	General information about the account (real name, phone number, and so on—also known as gecos)
unixUserPassword	No need to set this; not used in UNIX identity lookups for NAS authentication. Setting this puts the configured unixUserPassword value in plaintext.
unixHomeDirectory	Defines path to UNIX home directories when a user authenticates against LDAP from a Linux client. Set this if you want to use LDAP for UNIX home directory functionality.
loginShell	Defines path to the bash/profile shell for Linux clients when a user authenticates against LDAP.

*Denotes attribute is required for proper functionality with Cloud Volumes Service. Remaining attributes are for client-side use only.

Attribute	What it does
cn*	Specifies the UNIX group name. When using Active Directory for LDAP, this is set when the object is first created, but it can be changed later. This name cannot be the same as other objects. For instance, if your UNIX user named user1 belongs to a group named user1 on your Linux client, Windows doesn't allow two objects with the same cn attribute. To work around this, rename the Windows user to a unique name (such as user1-UNIX); LDAP in Cloud Volumes Service uses the uid attribute for UNIX user names.
gidNumber*	Specifies the UNIX group numeric ID.
objectClass*	Specifies what type of object is being used; Cloud Volumes Service requires group to be included in the list of object classes (this attribute is included in most Active Directory deployments by default).
memberUid	Specifies which UNIX users are members of the UNIX group. With Active Directory LDAP in Cloud Volumes Service, this field is not necessary. The Cloud Volumes Service LDAP schema uses the Member field for group memberships.
Member*	Required for group memberships/secondary UNIX groups. This field is populated by adding Windows users to Windows groups. However, if the Windows groups don't have UNIX attributes populated, they are not included in the UNIX user's group membership lists. Any groups that need to be available in NFS must populate the required UNIX group attributes listed in this table.

*Denotes attribute is required for proper functionality with Cloud Volumes Service. Remaining attributes are for client-side use only.

LDAP bind information

To query users in LDAP, Cloud Volumes Service must bind (login) to the LDAP service. This login has read-only permissions and is used to query LDAP UNIX attributes for directory lookups. Currently, LDAP binds are possible only by using an SMB machine account.

You can only enable LDAP for `CVS-Performance` instances and use it for NFSv3, NFSv4.1, or dual-protocol volumes. An Active Directory connection must be established in the same region as the Cloud Volumes Service volume for successful deployment of the LDAP-enabled volume.

When LDAP is enabled, the following occurs in specific scenarios.

- If only NFSv3 or NFSv4.1 is used for the Cloud Volumes Service project, then a new machine account is created in the Active Directory domain controller, and the LDAP client in Cloud Volumes Service binds to Active Directory by using the machine account credentials. No SMB shares are created for the NFS volume and default hidden administrative shares (see the section "Default hidden shares") have share ACLs removed.

- If dual-protocol volumes are used for the Cloud Volumes Service project, then only the single machine account created for SMB access is used to bind the LDAP client in Cloud Volumes Service to Active Directory. No additional machine accounts are created.
- If dedicated SMB volumes are created separately (either before or after NFS volumes with LDAP are enabled), then the machine account for LDAP binds is shared with the SMB machine account.
- If NFS Kerberos is also enabled, two machine accounts are created—one for SMB shares and/or LDAP binds and one for NFS Kerberos authentication.

LDAP queries

Although LDAP binds are encrypted, LDAP queries are passed over the wire in plaintext by using the common LDAP port 389. This well-known port cannot currently be changed in Cloud Volumes Service. As a result, someone with access to packet sniffing in the network can see user and group names, numeric IDs, and group memberships.

However, Google Cloud VMs cannot sniff other VM's unicast traffic. Only VMs actively participating in LDAP traffic (that is, being able to bind) can see traffic from the LDAP server. For more information about packet sniffing in Cloud Volumes Service, see the section "Packet sniffing/trace considerations."

LDAP client configuration defaults

When LDAP is enabled in a Cloud Volumes Service instance, an LDAP client configuration is created with specific configuration details by default. In some cases, options either do not apply to Cloud Volumes Service (not supported) or are not configurable.

LDAP client option	What it does	Default value	Can change?
LDAP Server List	Sets LDAP server names or IP addresses to use for queries. This is not used for Cloud Volumes Service. Instead, Active Directory Domain is used to define LDAP servers.	Not set	No
Active Directory Domain	Sets the Active Directory Domain to use for LDAP queries. Cloud Volumes Service leverages SRV records for LDAP in DNS to find LDAP servers in the domain.	Set to the Active Directory domain specified in the Active Directory connection.	No
Preferred Active Directory Servers	Sets the preferred Active Directory servers to use for LDAP. Not supported by Cloud Volumes Service. Instead, use Active Directory sites to control LDAP server selection.	Not set.	No

LDAP client option	What it does	Default value	Can change?
Bind using SMB Server Credentials	Binds to LDAP by using the SMB machine account. Currently, the only supported LDAP bind method in Cloud Volumes Service.	True	No
Schema Template	The schema template used for LDAP queries.	MS-AD-BIS	No
LDAP Server Port	The port number used for LDAP queries. Cloud Volumes Service currently uses only the standard LDAP port 389. LDAPS/port 636 is not currently supported.	389	No
Is LDAPS Enabled	Controls whether LDAP over Secure Sockets Layer (SSL) is used for queries and binds. Currently not supported by Cloud Volumes Service.	False	No
Query Timeout (sec)	Timeout for queries. If queries take longer than the specified value, queries fail.	3	No
Minimum Bind Authentication Level	The minimum supported bind level. Because Cloud Volumes Service uses machine accounts for LDAP binds and Active Directory does not support anonymous binds by default, this option does not come into play for security.	Anonymous	No
Bind DN	The user/distinguished name (DN) used for binds when simple bind is used. Cloud Volumes Service uses machine accounts for LDAP binds and does not currently support simple bind authentication.	Not set	No

LDAP client option	What it does	Default value	Can change?
Base DN	The base DN used for LDAP searches.	The Windows domain use for the Active Directory connection, in DN format (that is, DC=domain, DC=local).	No
Base search scope	The search scope for base DN searches. Values can include base, onelevel, or subtree. Cloud Volumes Service only supports subtree searches.	Subtree	No
User DN	Defines the DN where user searches start for LDAP queries. Currently not supported for Cloud Volumes Service, so all user searches start at the base DN.	Not set	No
User search scope	The search scope for user DN searches. Values can include base, onelevel, or subtree. Cloud Volumes Service does not support setting the user search scope.	Subtree	No
Group DN	Defines the DN where group searches start for LDAP queries. Currently not supported for Cloud Volumes Service, so all group searches start at the base DN.	Not set	No
Group search scope	The search scope for group DN searches. Values can include base, onelevel, or subtree. Cloud Volumes Service does not support setting the group search scope.	Subtree	No
Netgroup DN	Defines the DN where netgroup searches start for LDAP queries. Currently not supported for Cloud Volumes Service, so all netgroup searches start at the base DN.	Not set	No

LDAP client option	What it does	Default value	Can change?
Netgroup search scope	The search scope for netgroup DN searches. Values can include base, onelevel, or subtree. Cloud Volumes Service does not support setting the netgroup search scope.	Subtree	No
Use start_tls over LDAP	Leverages Start TLS for certificate based LDAP connections over port 389. Currently not supported by Cloud Volumes Service.	False	No
Enable netgroup-by-host lookup	Enables netgroup lookups by hostname rather than expanding netgroups to list all members. Currently not supported by Cloud Volumes Service.	False	No
Netgroup-by-host DN	Defines the DN where netgroup-by-host searches start for LDAP queries. Netgroup-by-host is currently not supported for Cloud Volumes Service.	Not set	No
Netgroup-by-host search scope	The search scope for netgroup-by-host DN searches. Values can include base, onelevel or subtree. Netgroup-by-host is currently not supported for Cloud Volumes Service.	Subtree	No
Client session security	Defines what level of session security is used by LDAP (sign, seal, or none). LDAP signing is supported by CVS-Performance, if requested by Active Directory. CVS-SW does not support LDAP signing. For both service types, sealing is currently not supported.	None	No

LDAP client option	What it does	Default value	Can change?
LDAP referral chasing	When using multiple LDAP servers, referral chasing allows the client to refer to other LDAP servers in the list when an entry is not found in the first server. This is currently not supported by Cloud Volumes Service.	False	No
Group membership filter	Provides a custom LDAP search filter to be used when looking up group membership from an LDAP server. Not currently supported with Cloud Volumes Service.	Not set	No

Using LDAP for asymmetric name mapping

Cloud Volumes Service, by default, maps Windows users and UNIX users with identical usernames bidirectionally without special configuration. As long as Cloud Volumes Service can find a valid UNIX user (with LDAP), then 1:1 name mapping occurs. For instance, if Windows user `johnsmith` is used, then, if Cloud Volumes Service can find a UNIX user named `johnsmith` in LDAP, name mapping succeeds for that user, all files/folders created by `johnsmith` show the correct user ownership, and all ACLs affecting `johnsmith` are honored regardless of the NAS protocol in use. This is known as symmetric name mapping.

Asymmetric name mapping is when the Windows user and UNIX user identity don't match. For instance, if Windows user `johnsmith` has a UNIX identity of `jsmith`, Cloud Volumes Service needs a way to be told about the variation. Because Cloud Volumes Service currently doesn't support creation of static name mapping rules, LDAP must be used to look up the identity of the users for both Windows and UNIX identities to ensure proper ownership of files and folders and expected permissions.

By default, Cloud Volumes Service includes `LDAP` in the ns-switch of the instance for the name map database, so that to provide name mapping functionality by using LDAP for asymmetric names, you only need to modify some of the user/group attributes to reflect what Cloud Volumes Service looks for.

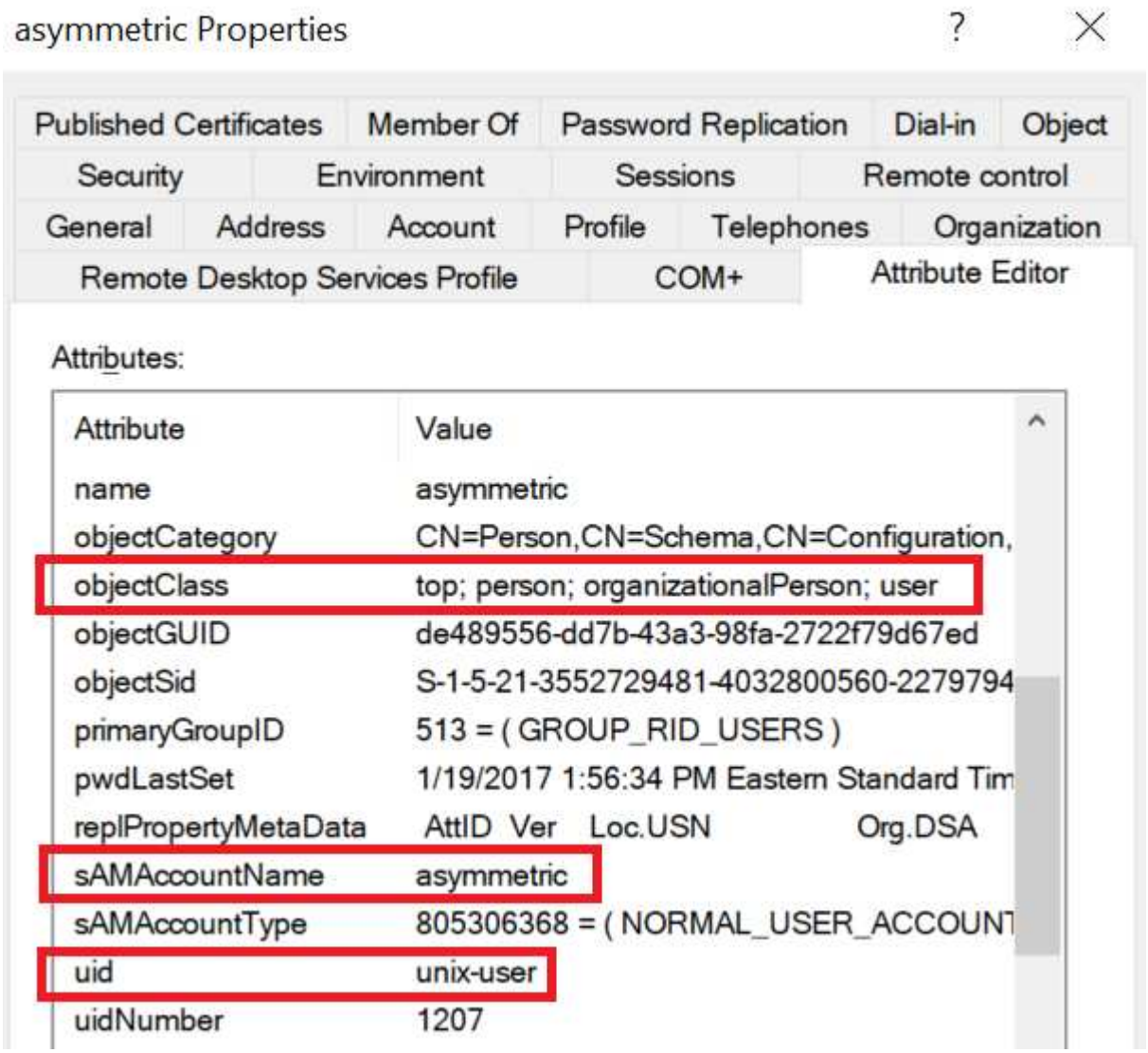
The following table shows what attributes must be populated in LDAP for asymmetric name mapping functionality. In most cases, Active Directory is already configured to do this.

Cloud Volumes Service attribute	What it does	Value used by Cloud Volumes Service for name mapping
Windows to UNIX objectClass	Specifies the type of object being used. (That is, user, group, posixAccount, and so on)	Must include user (can contain multiple other values, if desired.)
Windows to UNIX attribute	that defines the Windows username at creation. Cloud Volumes Service uses this for Windows to UNIX lookups.	No change needed here; sAMAccountName is the same as the Windows login name.
UID	Defines the UNIX username.	Desired UNIX username.

Cloud Volumes Service currently does not use domain prefixes in LDAP lookups, so multiple domain LDAP environments do not function properly with LDAP namemap lookups.

The following example shows a user with the Windows name `asymmetric`, the UNIX name `unix-user`, and the behavior it follows when writing files from both SMB and NFS.

The following figure shows how LDAP attributes look from the Windows server.



From an NFS client, you can query the UNIX name but not the Windows name:

```
# id unix-user
uid=1207(unix-user) gid=1220(sharedgroup) groups=1220(sharedgroup)
# id asymmetric
id: asymmetric: no such user
```

When a file is written from NFS as `unix-user`, the following is the result from the NFS client:

```
sh-4.2$ pwd
/mnt/home/ntfssh-4.2$ touch unix-user-file
sh-4.2$ ls -la | grep unix-user
-rwx------  1 unix-user sharedgroup     0 Feb 28 12:37 unix-user-nfs
sh-4.2$ id
uid=1207(unix-user) gid=1220(sharedgroup) groups=1220(sharedgroup)
```

From a Windows client, you can see that the owner of the file is set to the proper Windows user:

```
PS C:\ > Get-Acl \\demo\home\ntfs\unix-user-nfs | select Owner
Owner
-----
NTAP\asymmetric
```

Conversely, files created by the Windows user `asymmetric` from an SMB client show the proper UNIX owner, as shown in the following text.

SMB:

```
PS Z:\ntfs> echo TEXT > asymmetric-user-smb.txt
```

NFS:

```
sh-4.2$ ls -la | grep asymmetric-user-smb.txt
-rwx------  1 unix-user         sharedgroup   14 Feb 28 12:43 asymmetric-
user-smb.txt
sh-4.2$ cat asymmetric-user-smb.txt
TEXT
```

LDAP channel binding

Because of a vulnerability with Windows Active Directory domain controllers, Microsoft Security Advisory ADV190023 changes how DCs allow LDAP binds.

The impact for Cloud Volumes Service is the same as for any LDAP client. Cloud Volumes Service does not currently support channel binding. Because Cloud Volumes Service supports LDAP signing by default through negotiation, LDAP channel binding should not be an issue. If you do have issues binding to LDAP with channel binding enabled, follow the remediation steps in ADV190023 to allow LDAP binds from Cloud Volumes Service to succeed.

DNS

Active Directory and Kerberos both have dependencies on DNS for host name to IP/IP to host name resolution. DNS requires port 53 to be open. Cloud Volumes Service does not make any modifications to DNS records, nor does it currently support the use of dynamic DNS on network interfaces.

You can configure Active Directory DNS to restrict which servers can update DNS records. For more information, see Secure Windows DNS.

Note that resources within a Google project default to using Google Cloud DNS, which isn't connected with Active Directory DNS. Clients using Cloud DNS cannot resolve UNC paths returned by Cloud Volumes Service. Windows clients joined to the Active Directory domain are configured to use Active Directory DNS and can resolve such UNC paths.

To join a client to Active Directory, you must configure its DNS configuration to use Active Directory DNS. Optionally, you can configure Cloud DNS to forward requests to Active Directory DNS. See Why can't my client resolve the SMB NetBIOS name? for more information.

Cloud Volumes Service does not currently support DNSSEC and DNS queries are performed in plaintext.

File access auditing

Currently not supported for Cloud Volumes Service.

Antivirus protection

You must perform antivirus scanning in Cloud Volumes Service at the client to a NAS share. There is currently no native antivirus integration with Cloud Volumes Service.

Next: Service operation.

Copyright information

Trademark information