COHESITY

Version 1.0
December 2019

Cohesity DataPlatform: Identity and Access Management for File Services

Cohesity DataPlatform Multiprotocol Internals

ABSTRACT

This white paper describes identity and access management for file services on Cohesity DataPlatform by looking at Cohesity's integration with Active Directory and/or LDAP, and explains Cohesity's security styles (Native, Unified, and NTFS) and their impact on multiprotocol permissions. Ultimately, to help demystify how access works in a Cohesity DataPlatform multiprotocol environment.

Table of Contents

Figures

Tables

1 Introduction

Cohesity file services differentiates itself from traditional scale-out NAS as a next-gen product category. As a web-scale platform, this solution offers multiple benefits, including unlimited scale-out and unparalleled storage efficiency, with global variable-length block deduplication and compression across the cluster.

This white paper provides design considerations for understanding, configuring, and troubleshooting user access and file management with Cohesity DataPlatform™. Cohesity DataPlatform supports several protocols — like NFS, SMB, and S3 — which is why it's important to understand how identity management, ID mappings, and security styles interact in the context of multiprotocol support. The goal of this paper is to provide an understanding of Cohesity's implementation of multiprotocol support.

2 Multiprotocol Implementation

Unlike some NAS solutions that are implemented using Samba or NFS-Ganesha, Cohesity DataPlatform built support for SMB/CIFS, NFS, and S3 protocols from the ground up.

Cohesity DataPlatform provides the ability to create file exports/shares that can be accessed via NFS or SMB/CIFS protocols with Unified Permissions and are called DataPlatform “Views.” These Views are members of a “Storage Domain,” which is a logical data pool with defined storage policies for efficiency (deduplication & compression), storage resiliency settings erasure coding (EC) and/or replication factor (RF), and security settings encryption, and cloud tiering. Multiprotocol access to the same data allows support of applications across all major enterprise operating systems, including Microsoft Windows, Linux, and S3 API.

3 Support for NAS Protocols

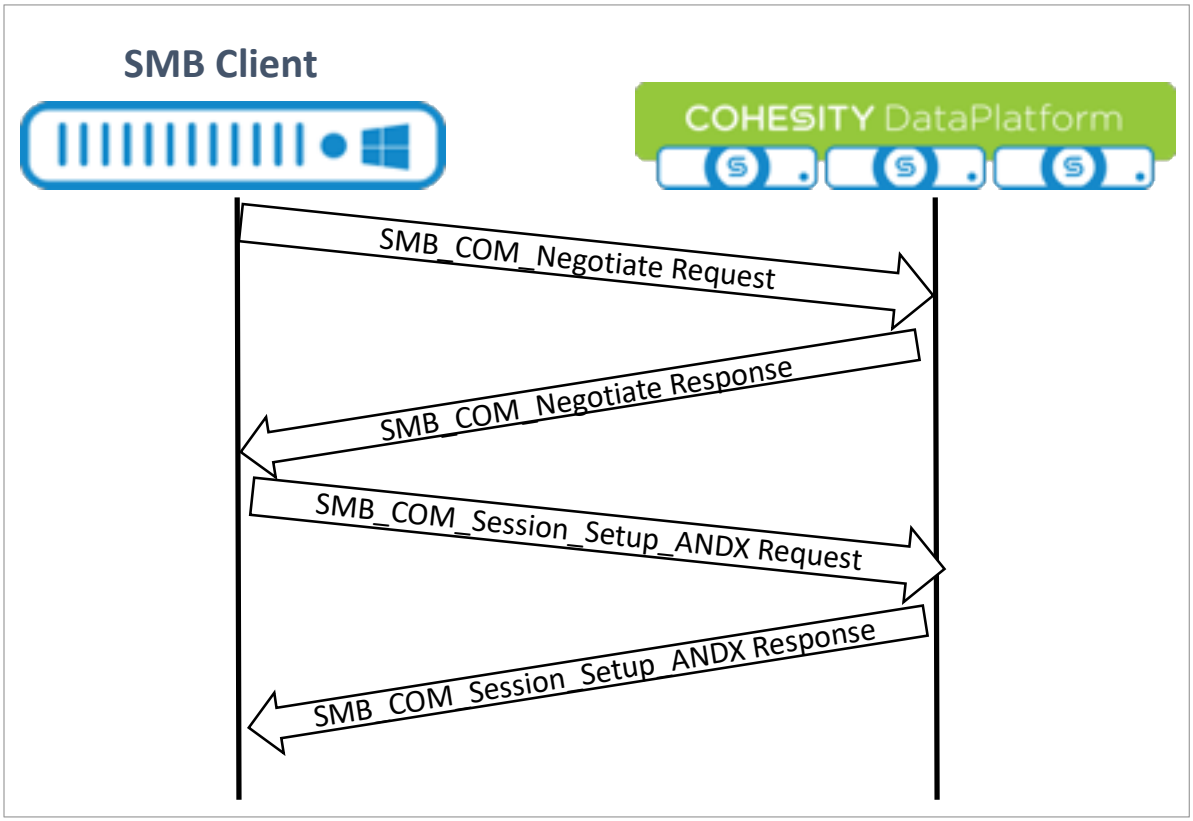
Cohesity DataPlatform supports the traditional NAS protocols, SMB/CIFS and NFS.

3.1 SMB Protocol

The Server Message Block (SMB) Protocol is a network file-sharing protocol. The SMB Protocol has grown over the years to accommodate the increasing functionality of Microsoft SMB Protocol. The set of message packets that defines a particular version of the protocol is called a “dialect.”

The dialect is determined during the SMB session setup.

Figure 1: SMB Session Setup



Cohesity DataPlatform supports the following SMB dialects and functionality:

Table 1: Supported SMB Dialects and Functionality

SUPPORTED SMB DIALECTS	SUPPORTED SMB FUNCTIONALITY
2.0	SMB Signing

SUPPORTED SMB DIALECTS
2.1
3.0
3.02

SUPPORTED SMB FUNCTIONALITY
SMB Encryption
Continuous Availability for SMB 2 & 3
Server-Side Copy
SMB VSS

For detailed information on the SMB protocol, see the Microsoft TechNet article, Microsoft SMB Protocol and CIFS Protocol Overview.

3.2 NFS Protocol

The Network File System (NFS) protocol allows systems to mount remote filesystems and use them as if they are local filesystems. Cohesity DataPlatform supports NFSv3, which is detailed in RFC1813. NFS is implemented using the RPC Protocol, designed to support remote procedure calls. All NFS operations are implemented as RPC procedures.

NFSv3 is a stateless protocol, which means the server does not need to maintain the state for any NFS clients.

4 Identity Management and Authentication

Before implementing file services, one needs to understand authentication. Authentication services verify a user's identity against a directory service before access to files and directories is granted. Cohesity DataPlatform integrates with several directory services, such as Active Directory and OpenLDAP.

4.1 AD (Active Directory) Integration

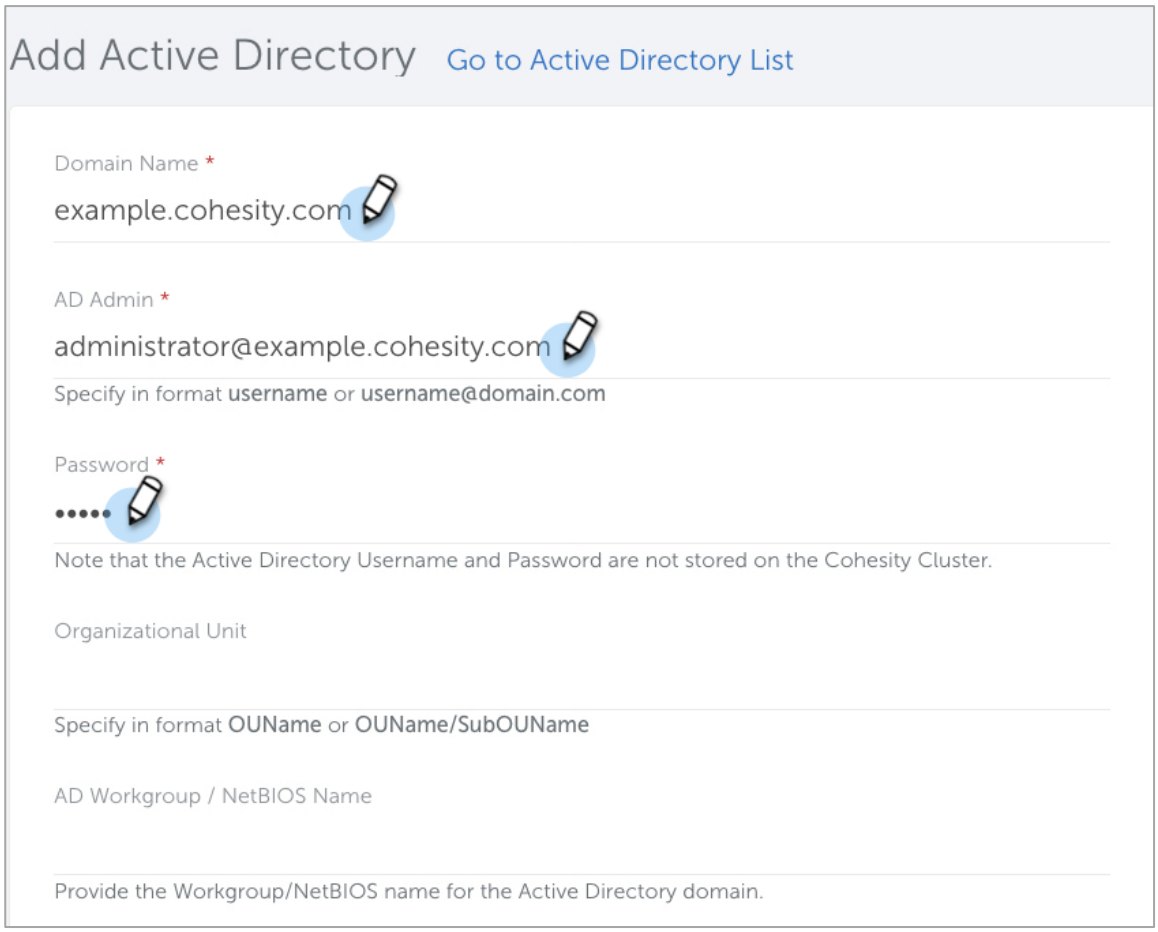
Active Directory is implemented by Microsoft and can provide several services: LDAP (Lightweight Directory Access Protocol), Kerberos, and DNS. Active Directory can provide user/group identity management and authentication for a Cohesity cluster. Active Directory can be used to authenticate all Windows clients and users. Cohesity DataPlatform is compatible with RFC2307. When deploying Cohesity DataPlatform in a multiprotocol environment, we recommend that you join the Cohesity cluster to Active Directory. RFC2307 allows you to associate Unix and Windows Active Directory accounts by adding attributes — such as a user ID (UID), group ID (GID), home directory, and shell — to an Active Directory object.

4.1.1 Configure Active Directory on Cohesity DataPlatform

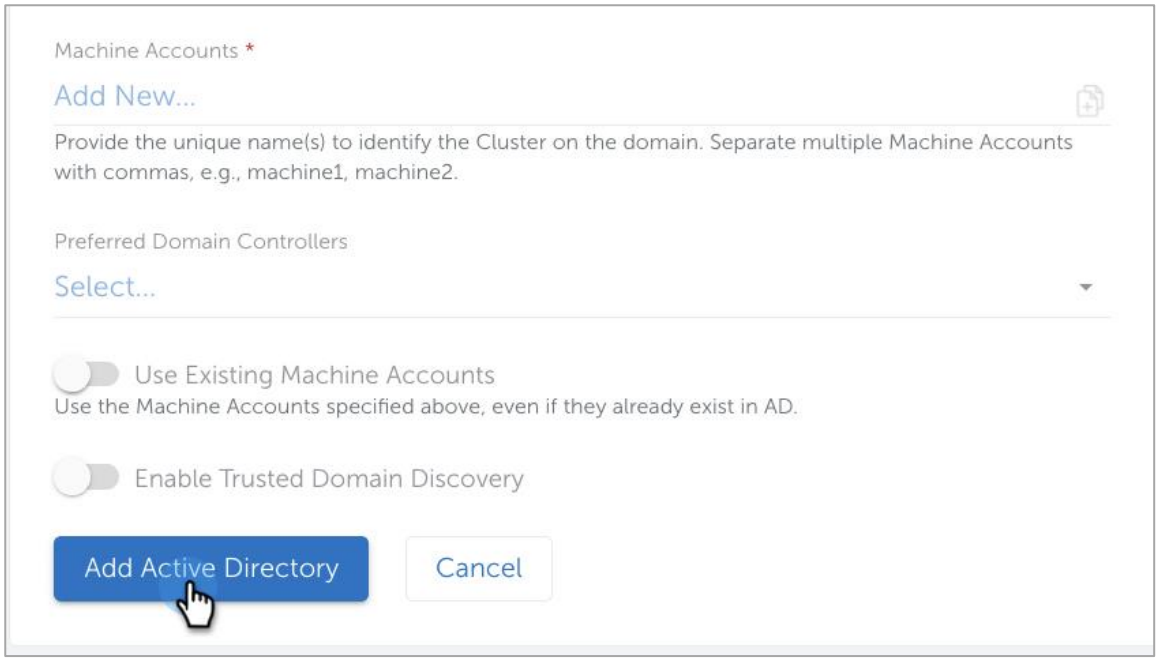
To connect Cohesity DataPlatform to Active Directory:

1. Log in to Cohesity DataPlatform.
2. Navigate to **Admin** > **Access Management**.
3. Click the **Active Directory** tab.
4. Click **Add Active Directory**.
5. Enter the **Active Directory Domain Name**.
6. Enter the **AD Admin** username and **Password** for a user with administrative rights to join computers to the Active Directory domain.

7. Optionally, enter a specific **Organizational Unit** where the computer account should be located in the Active Directory hierarchy.



8. The Cohesity cluster name is automatically listed as the machine account to be created. Optionally, in the same screen, you can provide **Machine Accounts** to identify the Cohesity cluster in Active Directory.

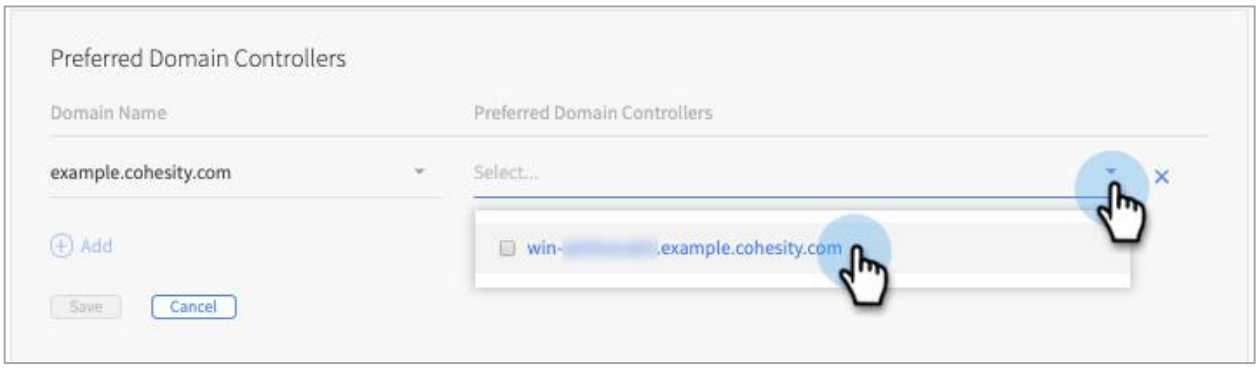


9. Click **Add Active Directory**.

4.1.2 Domain Controller Selection

As part of adding the cluster to Active Directory, the cluster can be configured to use a Preferred Domain Controllers for authentication. When you select the domain, a list of available domain controllers is presented. The cluster can be configured with multiple preferred domain controllers. When a Preferred Domain Controller(s) is configured, the cluster will only use those domain controllers for authentication.

Figure 2: Selecting Preferred Domain Controllers



If a preferred domain controller is not configured, the cluster performs queries to build a list of available domain controllers and attempts to contact the domain controllers. Based on those domain controller response queries, the cluster automatically selects which domain controllers to use for authentication.

4.2 LDAP (Lightweight Directory Access Protocol) Integration

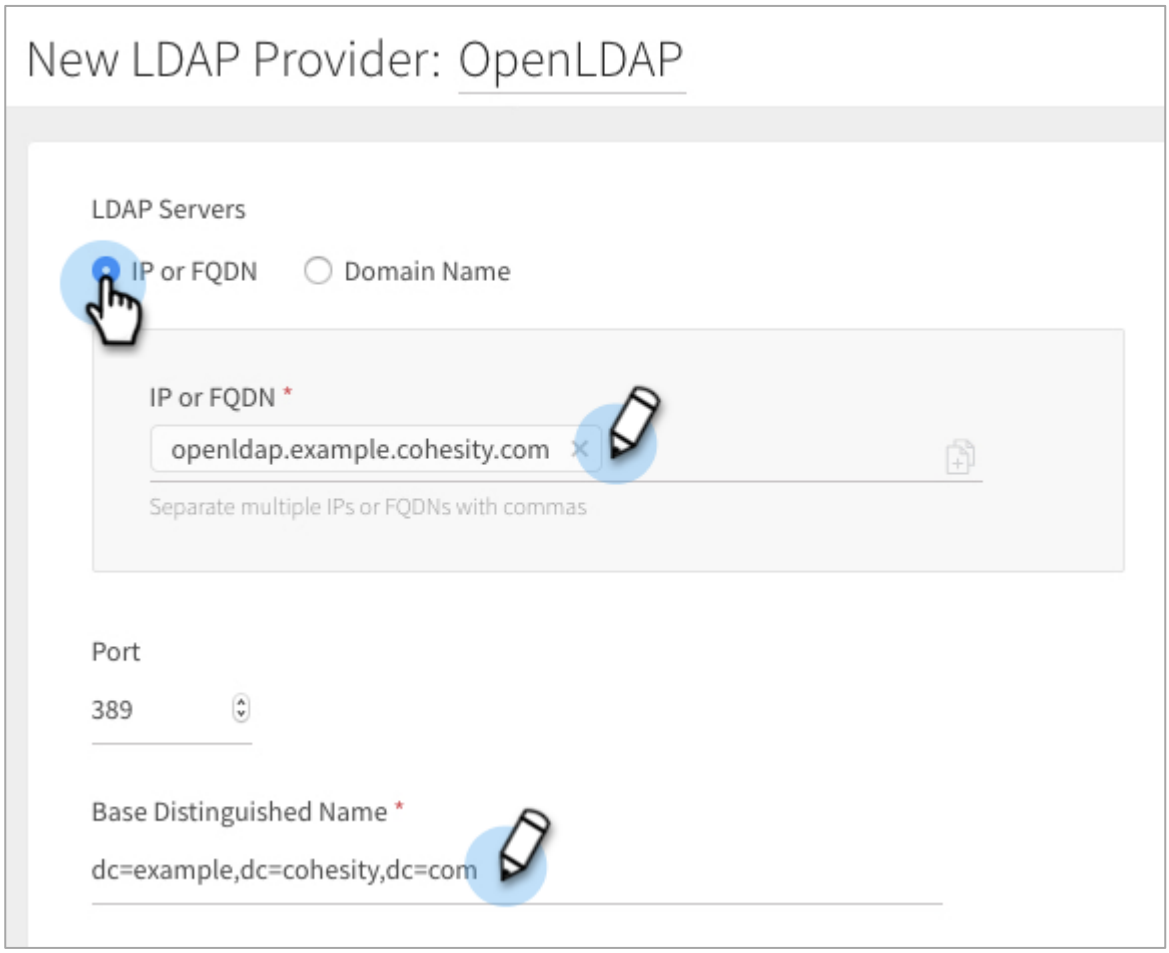
LDAP is an industry standard application protocol for accessing and maintaining distributed directory services. A common use of LDAP is to provide a central place to store usernames and passwords. This allows many different applications and services to use LDAP to connect to a directory server and validate users. A Cohesity cluster can be configured to use LDAP and can connect to LDAP Directory Servers like OpenLDAP.

4.2.1 Configure OpenLDAP on Cohesity DataPlatform

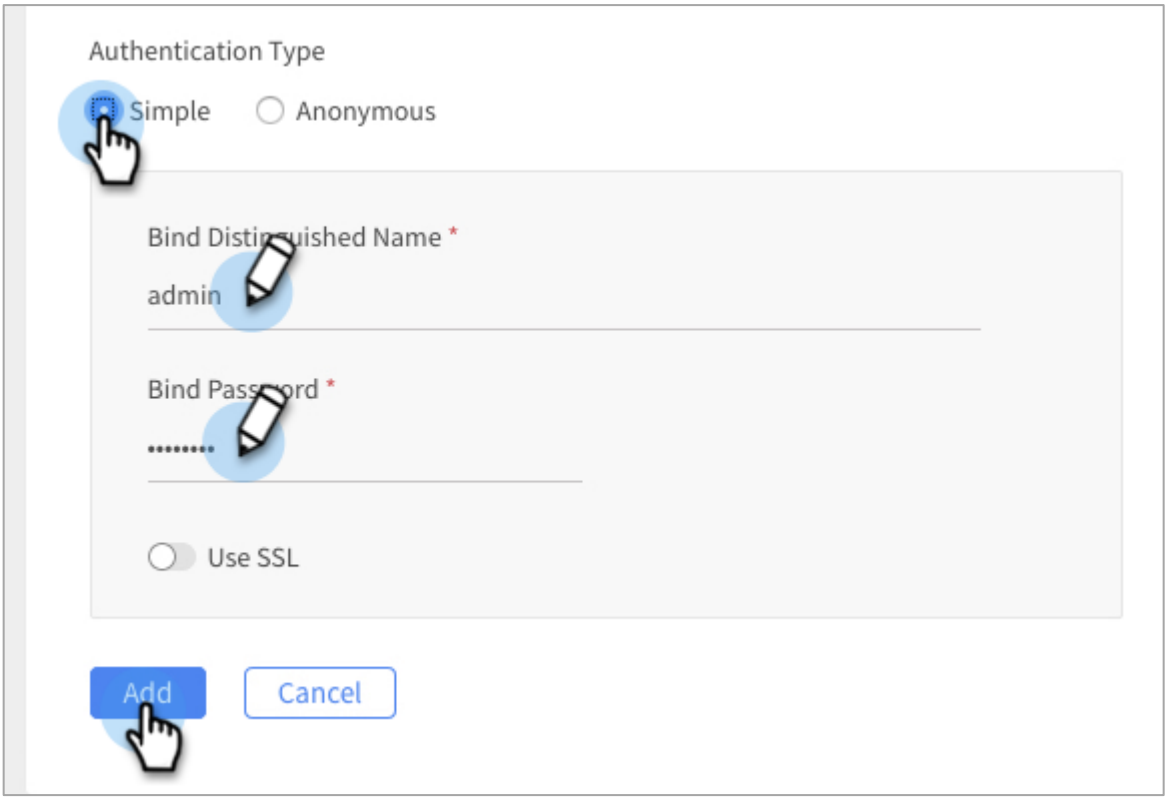
To connect Cohesity DataPlatform to OpenLDAP:

1. Log in to Cohesity DataPlatform.
2. Navigate to **Admin > Access Management**.
3. Click the **LDAP** tab.
4. Click **Add LDAP**.

5. Under **LDAP Servers**, enter the OpenLDAP **IP or FQDN**(s) and the **Base Distinguished Name**.



6. In the same screen, under **Authentication Type**, select **Simple** and enter the **Bind Distinguished Name** and **Bind Password**.

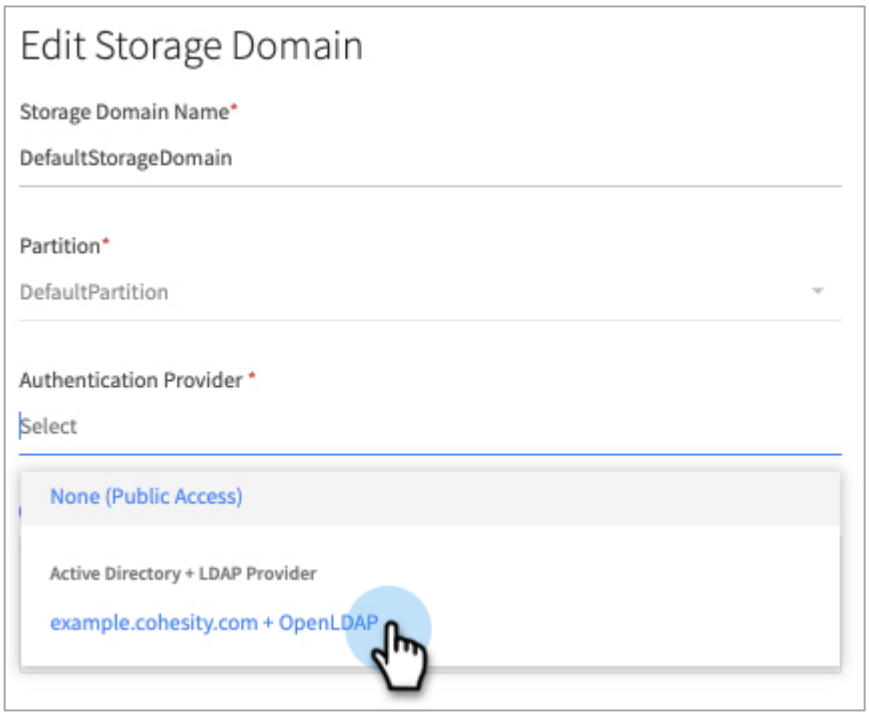


7. Click **Add**.

4.2.2 Storage Domain Authentication Provider Mapping

A Cohesity cluster can join multiple Active Directory and/or LDAP directory services. By default, a Storage Domain is accessible by all available authentication providers. A Storage Domain can be edited to use a specific Active Directory and LDAP mapping. Mapping to a specific Active Directory domain means that users from only that AD domain have permission to view the Storage Domain's contents when accessing through SMB.

Figure 3: Edit Storage Domain to Select Authentication Provider



4.2.3 Local Users and Groups

Cohesity DataPlatform supports using local authentication to access shares using NTLM (NT LAN Manager) authentication. Local users and groups can be configured and managed on the Cohesity cluster. Local groups can have Active Directory principals as a member.

5 Cohesity DataPlatform ID Mapping

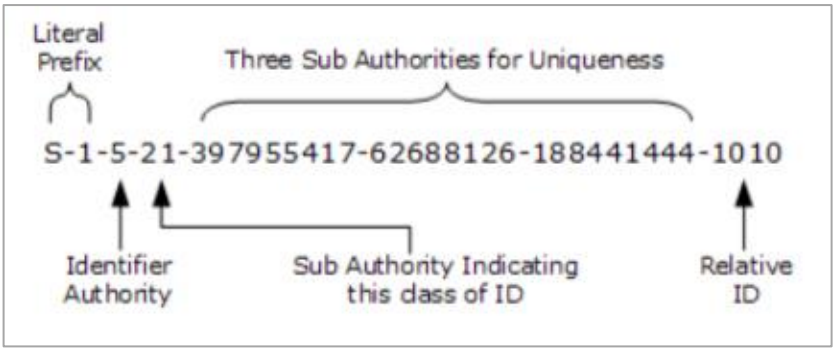
Windows and Unix have different methods of implementing access control.

In Unix, all users have a user ID and a group ID, and a unique numerical identification number called a userid (UID) and a groupid (GID). File, directory, and device (special file) permissions are granted based on "user," "group," or "other" (world) identification status. Permission is granted (or denied) for read, write, and execute access.

Windows grants or denies access and privileges to resources based on access control lists (ACLs), which use Security Identifiers (SIDs) to uniquely identify users and their group memberships. When a user logs in to a computer, an access token is generated that contains user and group SIDs, as well as the user privilege level. When a user requests access to a resource, the access token is checked against the ACL to permit or deny that particular action on that particular object.

ID Mapping is needed to map the SIDs of Windows users and SID of the group to the UIDs and GIDs of a Unix user and group. Cohesity offers five different methods of mapping users.

Table 2: ID Mapping Types

ID MAPPING TYPE	DESCRIPTION
Fixed	A user-defined UID and GID are applied to all files created on the View.
Relative ID of SID (RID)	RID is a variable-length number that is assigned to objects at creation and becomes part of the object's Security Identifier (SID) that uniquely identifies an account or group within a domain. For every Windows user or group, the RID of the corresponding SID is used as the UID or GID. 
RFC-2307	Active Directory has the UID and GID information. ▪ The UID for a user is stored in the attribute ‘uidNumber’. ▪ The GID for a group is stored in the attribute 'gidNumber'.

ID MAPPING TYPE	DESCRIPTION
Service for Unix (SFU) 3.0	Active Directory has the UID and GID information. ▪ The UID for a user is stored in the attribute ‘uidNumber’. ▪ The GID for a group is stored in the attribute 'gidNumber'
Centrify	Centrify stores information about ID mapping in Active Directory in their own format.

NOTE: ID mapping is used only for Views that support both the NFS (Unix) and SMB (Windows) protocols and have the Security Style set to either Unified or NTFS. ID mapping is not needed for Views that support just one protocol.

5.1 Configure RFC-2307 and Services for Unix (SFU)

In this guide, we detail the steps required to configure ID mapping for the RFC-2307 and SFU mapping types.

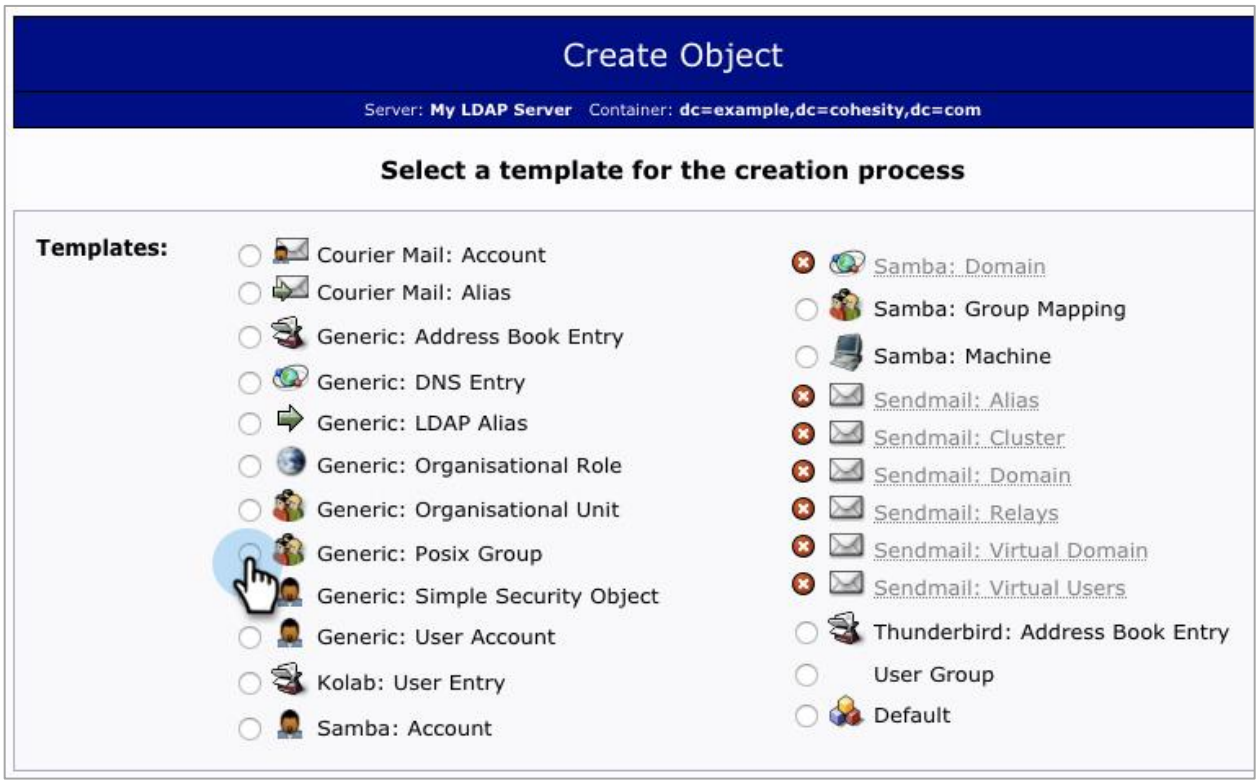
In this example, we have the following components:

- Microsoft Active Directory
- OpenLDAP 2.4.31
- phpLDAPadmin 1.2.2

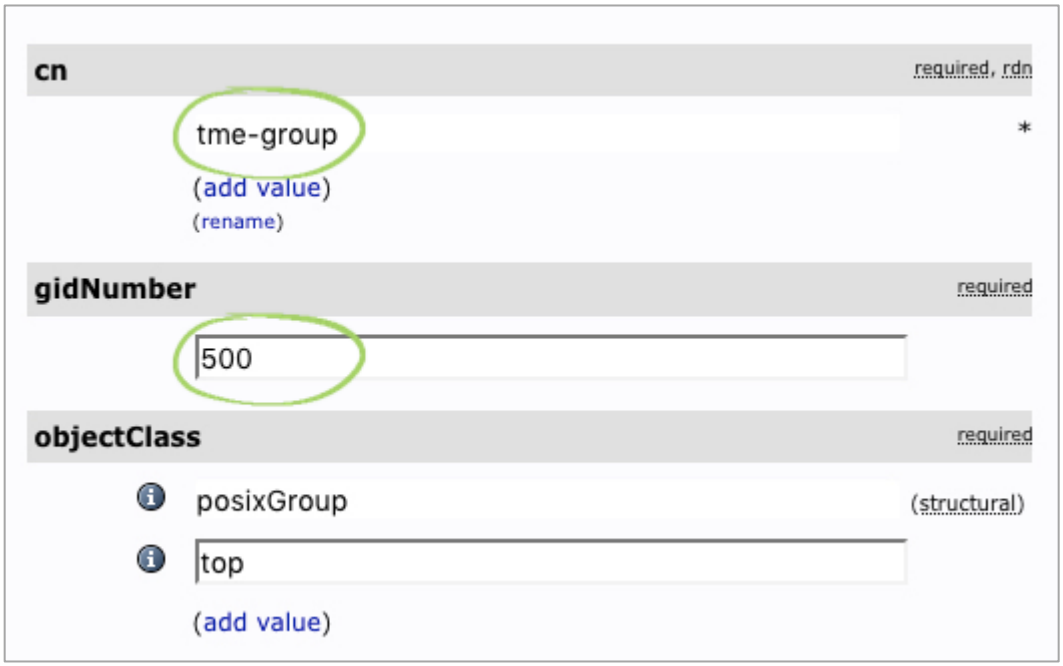
5.1.1 Create a Group in OpenLDAP Using phpLDAPadmin

The group you create will be used to map to the Active Directory group “Domain Users”.

Select **Generic: Posix Group**.



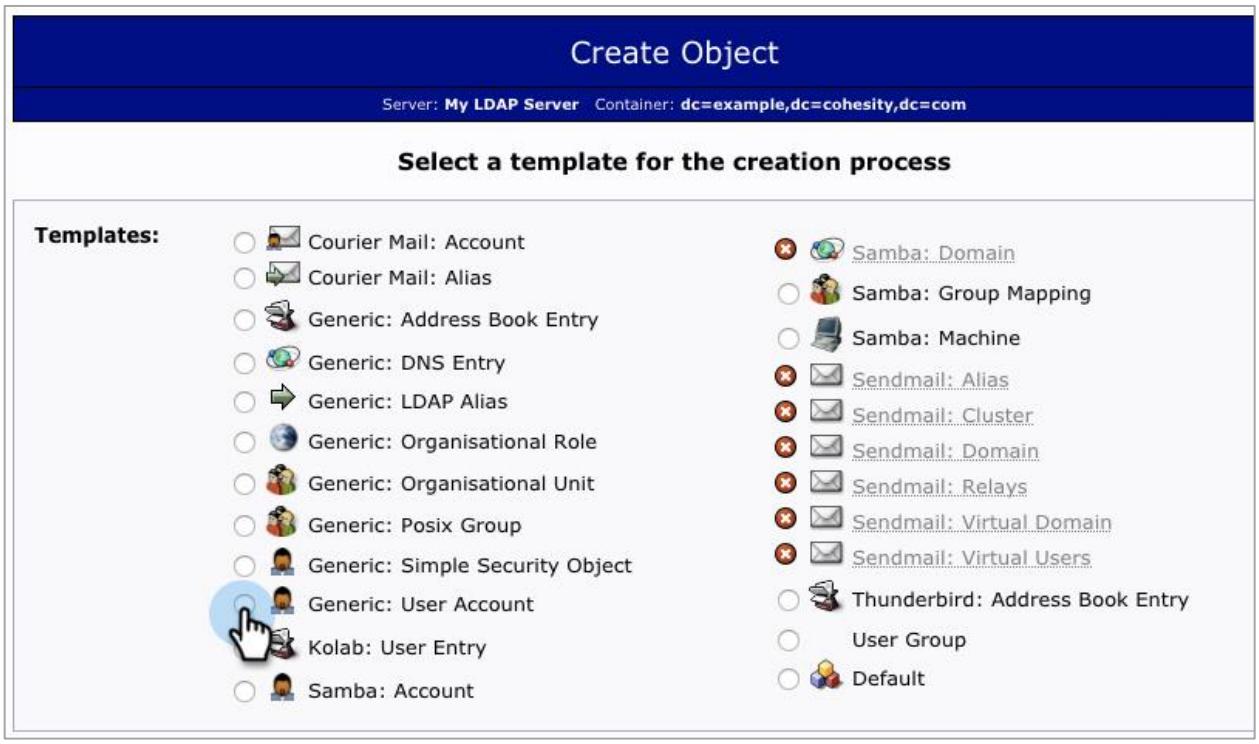
In this example, the group created is “tme-group”, with a gidNumber of 500.



5.1.2 Create a User in OpenLDAP Using phpLDAPadmin

Next, use phpLDAPadmin Create Object to create a user account. The user account is assigned to the group that was created in the previous step.

Select **Generic: User Account**.



In our example, we created a user with the username “tmeuser” and:

- The user’s default group is “tme-group”.
- The user’s home directory is /home/users/tmeuser.
- The user’s login shell is /bin/sh.
- The user has a uid of 2001.

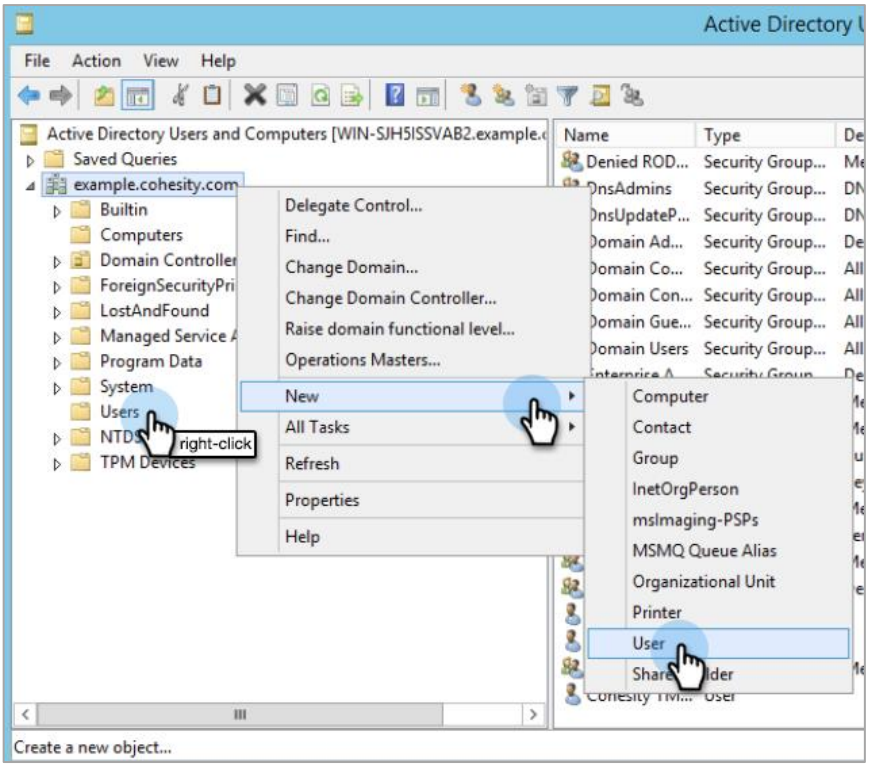
5.1.3 Create a User in Active Directory

Next, create the user in Active Directory who will be mapped to the user created in OpenLDAP.

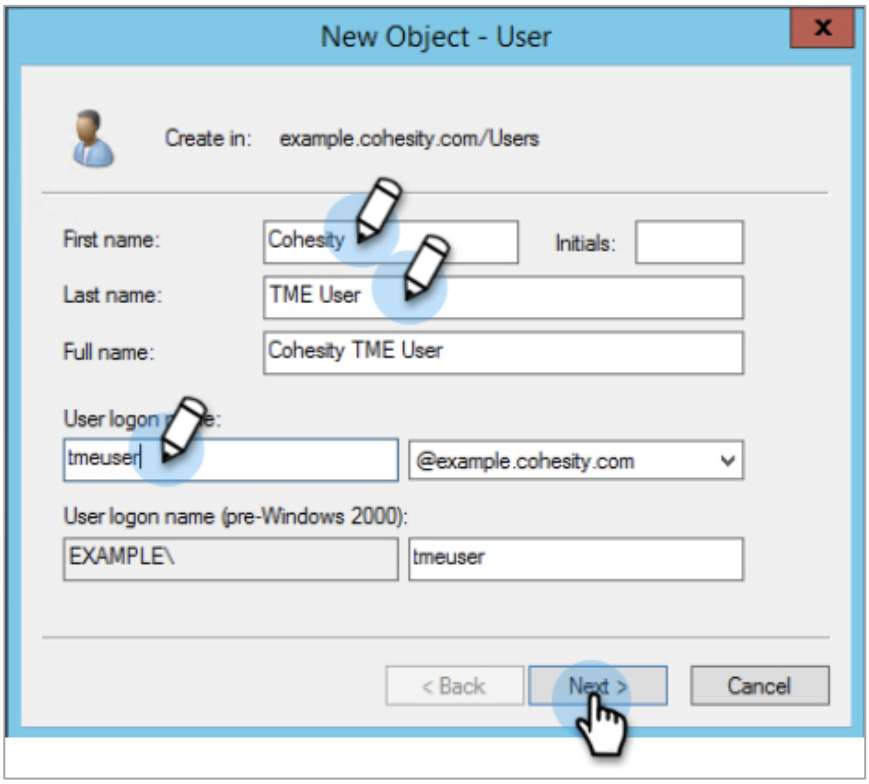
To create a new user in AD:

1. In Windows, select **Start** > **Administrative Tools** > **Active Directory Users and Computers** to open the Active Directory Users and Computers console.

2. Click the domain name that you created, and then expand the contents. Right-click **Users** and select **New** > **User**.



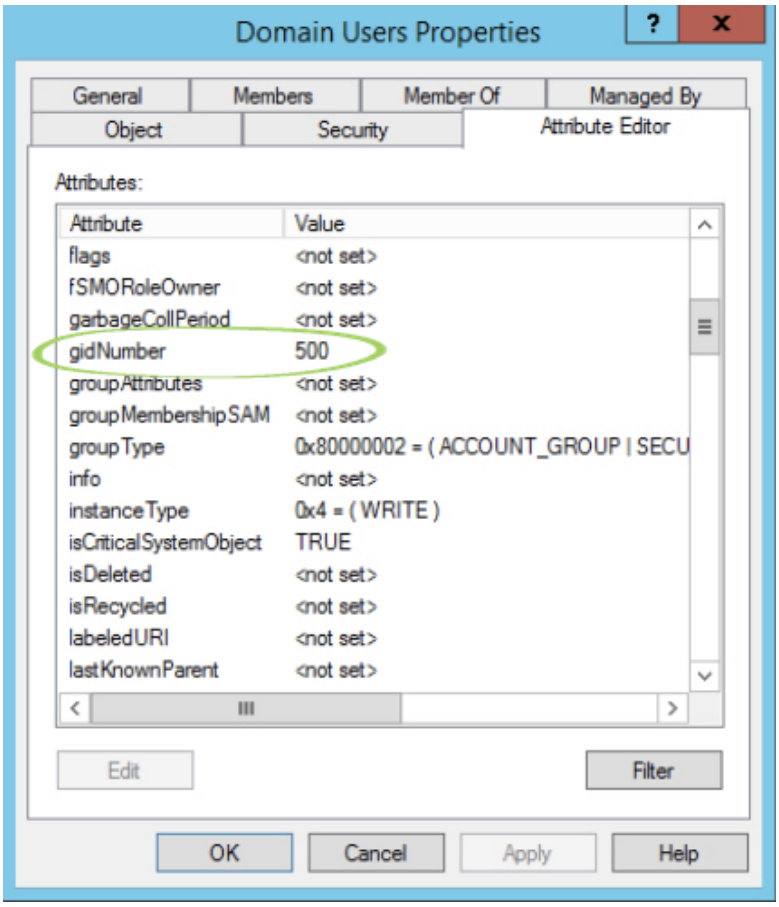
3. The **First name**, **Last name**, and **User logon name** of the new user, and then click **Next**.



4. Type a new **Password**, Confirm the password, and then select one of the following:
   - **Users must change password at next logon (recommended for most users)**
   - **User cannot change password**
   - **Password never expires**
   - Account is disabled
5. Click **Next**.
6. Review the information that you provided, and if everything is correct, click **Finish**.

5.1.4 Edit the Group Attributes in Active Directory

Using the Attribute Editor, modify “Domain Users” to update the gidNumber attribute to 500.



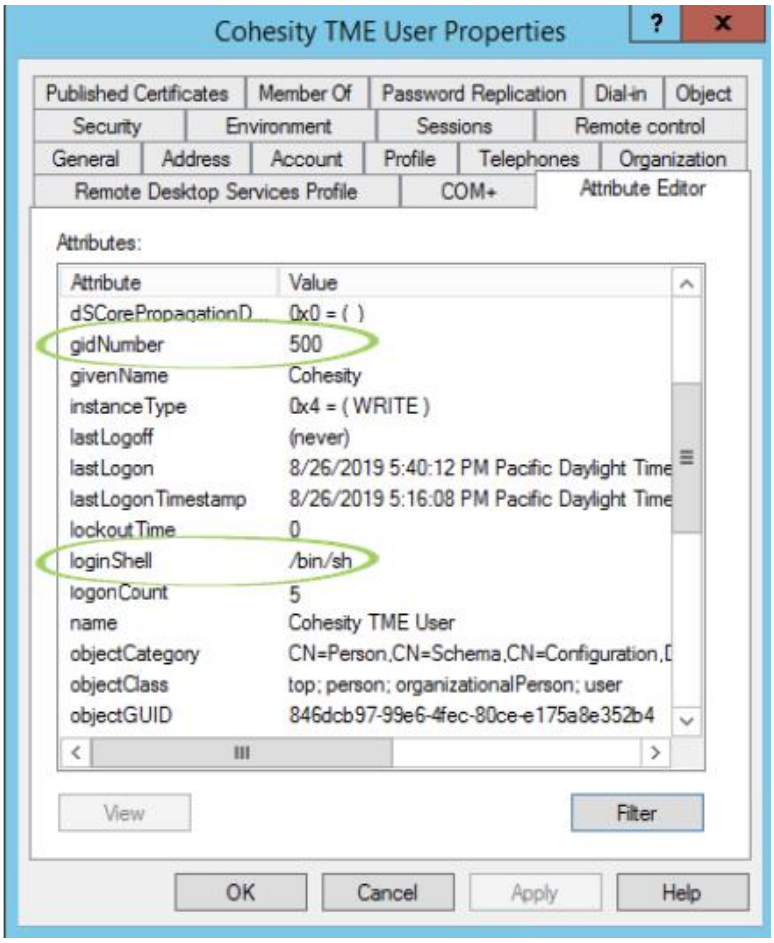
5.1.5 Edit the User Attributes in Active Directory

If you use Microsoft Active Directory with Windows Services for UNIX and RFC-2307 attributes to manage Linux, UNIX, and Windows systems, the following fields are required in Active Directory:

- uid
- uidNumber
- gidNumber
- loginShell
- UNIXHomeDirectory

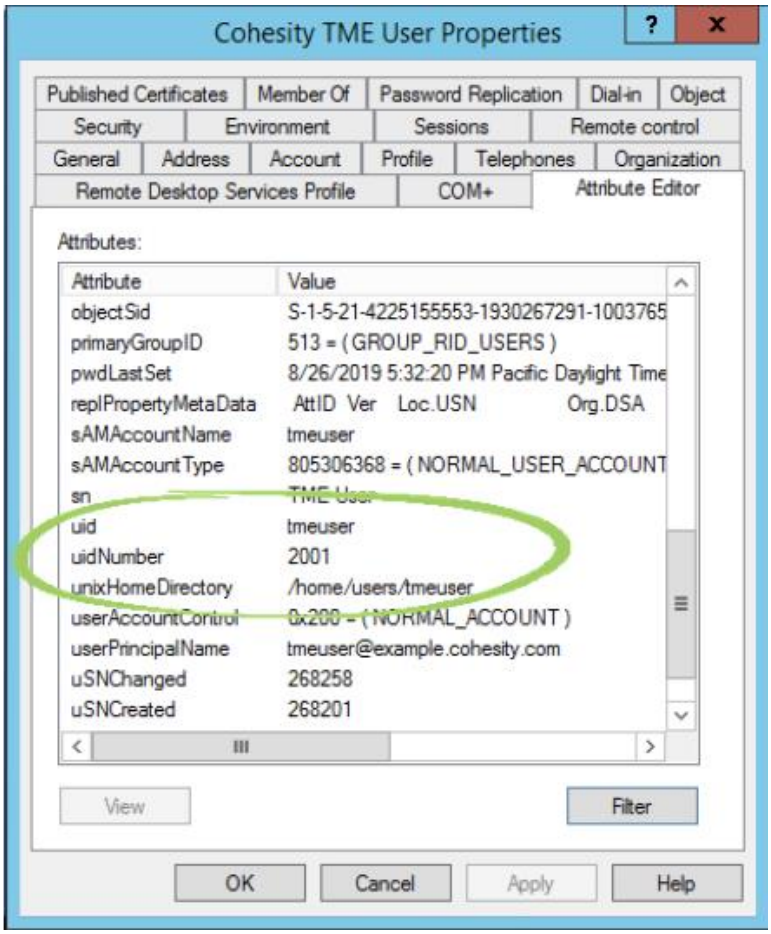
The following examples illustrate updating the attributes in Active Directory to match the values you set earlier.

- gidNumber = 500
- loginShell = /bin/sh



And:

- uid = tmeuser
- uidNumber = 2001
- unixHomeDirectory = /home/users/tmeuser



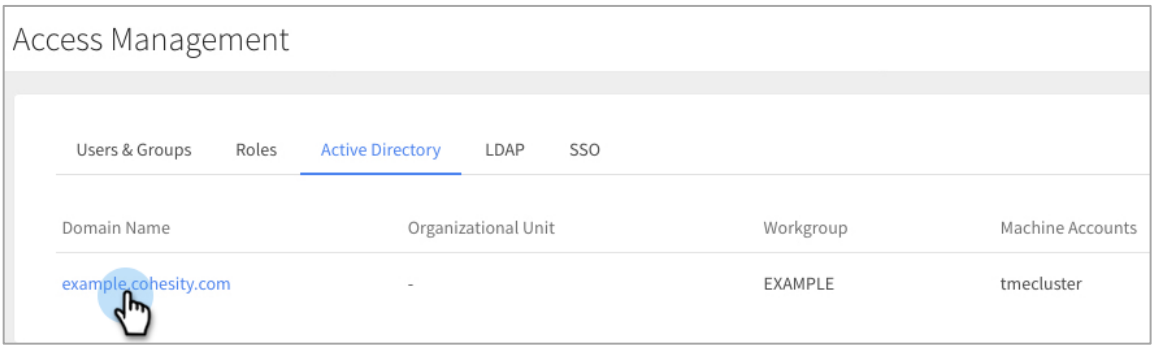
5.1.6 Configure the Cohesity Cluster for RFC-2307 and/ Services for Unix

In the following, we configure ID Mapping on the Cohesity Cluster for an environment where RFC2307 is deployed.

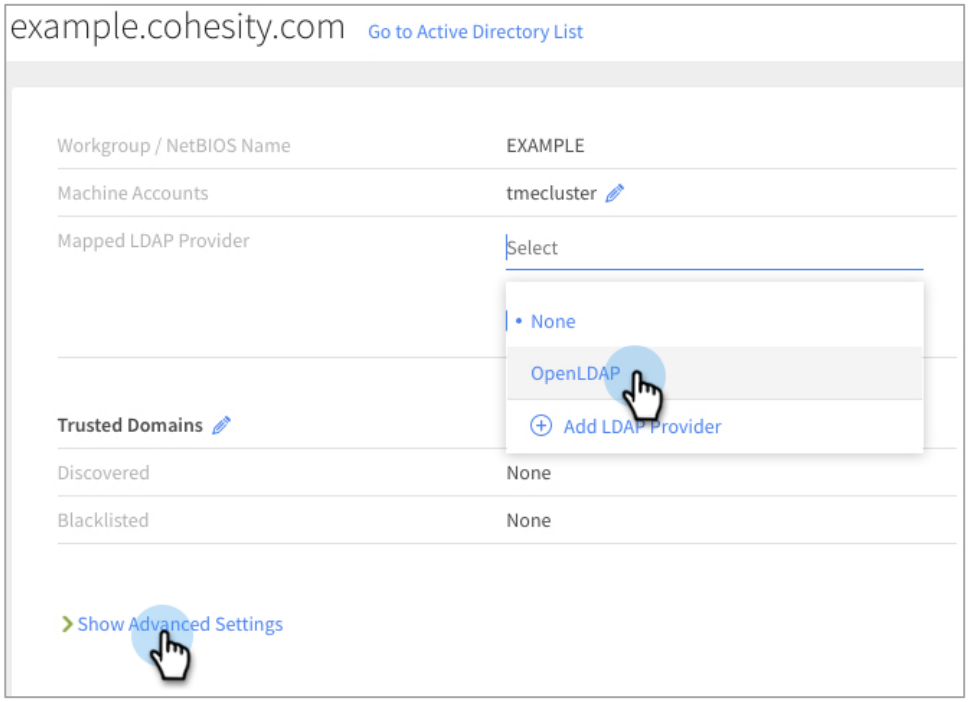
To configure your Cohesity cluster for RFC-2307 and Services for Unix:

1. Log in to Cohesity DataPlatform.
2. Navigate to **Admin > Access Management**., then click the **Active Directory** tab.

3. Click the **Domain Name** that needs to be mapped to LDAP.

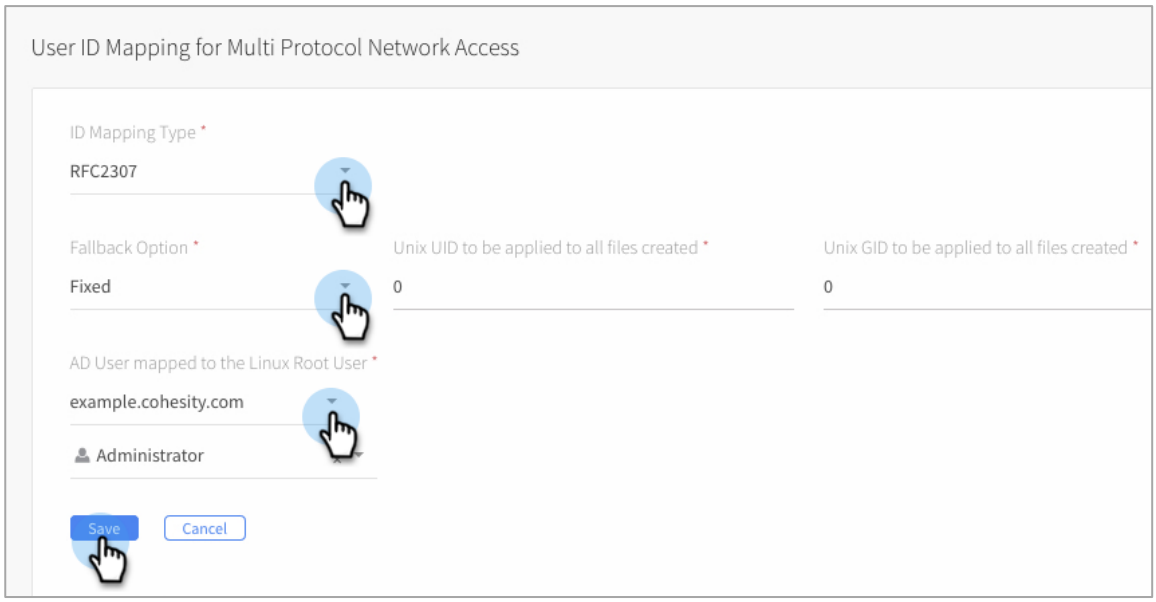


4. Under **Mapped LDAP Provider**, select the LDAP entry and then click **Show Advanced Settings**.



5. Click **Add ID Mapping**.

6. Under **ID Mapping Type**, choose **RFC2307**. Select a Fallback Option (the Unix UID and GID to set on a file when a user cannot be found). Enter the **AD User mapped to the Linux Root User**. Click **Save**.



6 Cohesity Security Styles

In a multiprotocol environment, a Cohesity View can be configured with any of three different security styles: Native, Unified, or NTFS.

Table 3: Cohesity Security Styles

SECURITY STYLES	DESCRIPTION	USE CASE RECOMMENDATION
Native	▪ Windows and Linux permissions are stored separately. ▪ If an NFS client sets or changes the permissions on a file, the CIFS/SMB permissions are not changed. ▪ If a CIFS/SMB client sets or changes the ACLs on a file, the NFS permissions are not changed.	Recommended where file access to the View is primarily via an application.
Unified	▪ Windows and Linux permissions kept consistent. ▪ Files created in SMB get the inherited ACLs from the parent and owner/group set to the user/user's group that created the file. ▪ Files created with uid and gid provided by the Unix machine. They will be appropriate ids based on the way the Unix machine is set up to get the id.	Recommended for environments where customers have users who are accessing shares/exports via both SMB and NFS.
NTFS	▪ SMB ACL (Access Control List) provides access to files and folders. ▪ In this type of security, file access via NFS and SMB is governed by the SMB ACL that is present on the file. ▪ Modifications to the SMB ACL via SMB will be applied to the ACL as if it was a regular SMB 'set info' operation. ▪ Modifications to NFS owner, group, and mode bits are not allowed.	Recommended for environments where files are accessed primarily via SMB.

Best Practice: Cohesity recommends a security policy of Unified for multiprotocol file services use cases.

7 Cohesity View Permissions

Cohesity DataPlatform provides multiple levels of permissions to access views that are exposed as SMB shares and/or NFS exports. Access to the views can be controlled via IP whitelists. View-level permissions are stored on Cohesity DataPlatform and provide an option for administrators to control access to the SMB shares / NFS exports and files within the view.

7.1 Whitelist Cohesity View

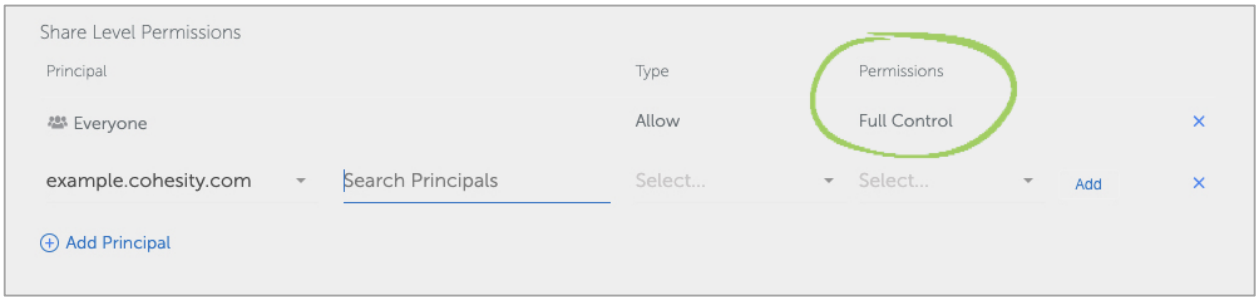
Whitelists allow the administrator to control which clients can access Cohesity Views.
For NFS, a whitelist is similar to the `/etc/exports` file in Unix.
Unlike a traditional `/etc/exports` file, the whitelist also controls which clients can access the View via SMB and S3. Whitelists can be set globally, per-View, and, starting in version 6.4, on a per-SMB share basis.

7.2 SMB Share Level Permissions

Starting with Cohesity DataPlatform version 6.4, Cohesity introduces Share Level Permissions. Share Level Permissions are independent of file/folder permissions and are set on the root View or a Share. Share Level Permissions work in conjunction with the NTFS file permissions. Access to a file is granted if and only if both the NTFS file permission and Share Level Permission allow it.

The following permission modes are supported for Share Level Permissions:

- **Full Control**. User has full control and can modify the permissions on the Share.
- **Modify**. User has permissions for CRUD files/folders in the share but cannot modify the access permissions.
- **Read Only**. User has read only access to the share.



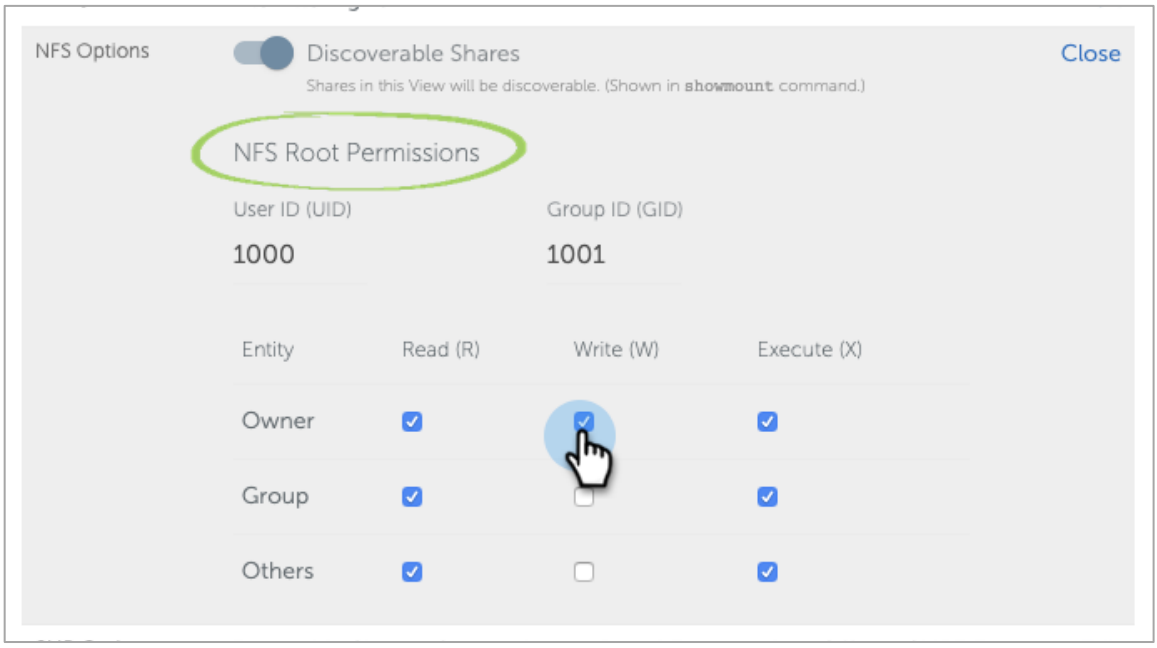
When both the Share Level Permissions and NTFS file permissions are set, the most restrictive access applies to the share and determines the effective access.

Table 4: NTFS vs Share Permissions — Most Restrictive Governs

NTFS FILE PERMISSION	SHARE PERMISSION	EFFECTIVE ACCESS
Full control	Read only	Read only
Read only	Full control	Read only
Full control	Full control	Full control

7.3 NFS Root Permissions

Starting with Cohesity DataPlatform version 6.4, a Cohesity administrator has the ability to change the NFS mode bits on a View and can set the UID/GID for the root of the View.



8 Appendix: Windows/Unix Posix Permissions Mapping

As part of Cohesity DataPlatform's unified permission model, Cohesity maps Windows ACLs to Posix mode bits, and vice-versa. The following tables show how the various ACLs and permissions map to their counterpart.

Table 5: Windows ACLs to Unix Posix Mode Bits

WINDOWS ACL	UNIX POSIX MODE BITS
FILE_ADD_FILE	d-w-
FILE_ADD_SUBDIRECTORY	d-w-
FILE_ALL_ACCESS	drwx or -rwx
FILE_APPEND_DATA	--w- or d-w-
FILE_DELETE_CHILD	d-w-
FILE_EXECUTE	---x
FILE_LIST_DIRECTORY	dr--
FILE_READ_ATTRIBUTES	-r--
FILE_READ_DATA	-r--
FILE_READ_EA	-r--
FILE_TRAVERSE	d--x
FILE_WRITE_ATTRIBUTES	--w-

WINDOWS ACL	UNIX POSIX MODE BITS
FILE_WRITE_DATA	--w-
FILE_WRITE_EA	--w-
DELETE	d-w- or --w-
READ_CONTROL	dr-- or -r--
WRITE_DAC	drwx or -rwx
WRITE_OWNER	drwx or -rwx
SYNCHRONIZE	NA

Table 6: Unix Posix Mode Bits to Windows ACLs

UNIX POSIX MODE BITS	WINDOWS ACL
Read	FILE_LIST_DIRECTORY, FILE_READ_ATTRIBUTES, FILE_READ_DATA, FILE_READ_EA
Write	FILE_ADD_FILE, FILE_WRITE_DATA, FILE_ADD_SUBDIRECTORY, FILE_APPEND_DATA, DELETE, FILE_DELETE_CHILD, FILE_WRITE_ATTRIBUTES, FILE_READ_EA
Execute	FILE_TRAVERSE/FILE_EXECUTE

9 About the Authors

Scott Owens is a Technical Marketing Engineer at Cohesity. In his role, Scott focuses on file services.

Other essential contributors included:

- Adaikkappan Arumugam, Sr Manager, Tech Marketing, Solutions Engineering & Tech Pubs
- Vibhor Gupta, Product Management File Services

10 Document Version History

VERSION	DATE	DOCUMENT HISTORY
1.0	Dec 2019	First full release

11 Your Feedback

Was this document helpful? Send us your feedback!

ABOUT COHESITY

Cohesity makes your data work for you by consolidating secondary storage silos onto a hyperconverged, web-scale data platform that spans both private and public clouds. Enterprise customers begin by radically streamlining their backup and data protection, then converge file and object services, test/dev instances, and analytic functions to provide a global data store. Cohesity counts many Global 1000 companies and federal agencies among its rapidly growing customer base and was named to Forbes' "Next Billion-Dollar Startups 2017," LinkedIn's "Startups: The 50 Industry Disruptors You Need to Know Now," and CRN's "2017 Emerging Vendors in Storage" lists.

For more information, visit our website and blog, follow us on Twitter and LinkedIn and like us on Facebook.

2000023-002-EN