COHESITY

Version 1.0

Dec 2021

Best Practices for Integrating SmartFiles with Active Directory

Generic Guidelines and Recommendations

ABSTRACT

This guide provides an overview of Cohesity's implementation of services for Active Directory integration and the best practices and recommendations for the Active Directory configuration.

Table of Contents

Tables

Cohesity SmartFiles

Cohesity SmartFiles delivers enterprise-class, unstructured data management with unified file and object storage and intelligent management applications such as security and tiering. As a web-scale platform, this solution offers multiple benefits, including unlimited scale-out, unparalleled storage efficiency, and robust fault tolerance across your Cohesity cluster.

Generic Recommendations for Active Directory Integration

You can join your Cohesity cluster to Active Directory (AD), the Microsoft directory service for Windows domain networks, for user authentication and authorization. A server running the Active Directory Domain Service (AD DS) role is called a domain controller. It authenticates and authorizes all users and machines in the Active Directory domain. Active Directory uses Lightweight Directory Access Protocol (LDAP) versions 2 and 3, and Microsoft's version of Kerberos.

The following sections include important considerations and recommendations while joining the Cohesity cluster to the Active Directory domains or creating the new Machine Account in a Cohesity cluster that's already joined to the Active Directory domains.

For details on some of the Active Directory features and services discussed in this section, see Appendix A—Brief Overview of Active Directory.

Domain Name Service (DNS)

A Cohesity cluster usually consists of multiple nodes. So, Cohesity highly recommends you spread all the SmartFiles workloads across all the nodes, ensuring consistency and availability. Cohesity recommends leveraging the Internal Load Balancer DNS service to balance the client connections across all the nodes. For more details and best practices on Cohesity's Internal Load Balancer, see SmartFiles Internal Load Balancing Guidelines.

While joining the Cohesity cluster or creating a new Machine Account, always specify the DNS hostname. This DNS hostname should be the Fully Qualified Domain Name (FQDN) for one of the DNS resolver VIPs. Below are some of the common scenarios in which DNS hostname declared in a Machine Account (Computer Object) are used:

- Managing Shares from Windows Microsoft Management Console (MMC)
- Disjoint DNS Namespace resolution during lookups

You can also easily add or modify the DNS hostname for an existing Machine Account from the cluster UI.

For detailed information on the configuration steps, see Appendix B—DNS IP Address for the Machine Account (AD Computer Object).

Time Synchronization

Always ensure that the difference in the computer clock between the Cohesity cluster and the domain controllers is less than the defined maximum time (default – 5 minutes). You can use NTP servers to synchronize the time and date between the Cohesity cluster and the domain controller.

Active Directory Authentication Protocols (Kerberos vs NTLM)

Kerberos is the default authentication protocol for Active Directory (Windows 2000 onwards) and Microsoft recommends using Kerberos over NTLM. NTLM (Windows NT LAN Manager) protocol has two versions: NTLMv1 and NTLMv2. NTLMv2 is more secure than NTLMv1.

NTLM authentication is only supported when the cluster joins only one Active Directory. Also, the supported NTLM version is determined by Active Directory, as illustrated in the following table.

Table 1: NTLM Supported Versions

IF AD SUPPORTS		THEN COHESITY SUPPORTS	
NTLMv2	**NTLMv1**	**NTLMv2**	**NTLMv1**
✔	✔	✔	✔
✔	✗	✔	✗
✗	✔	✗	✔

In case of multiple domains:

- NTLM is not supported when the Cohesity cluster is joined to multiple Active Directory domains (in single or multi-tenant).
- NTLM is supported when the Cohesity cluster is joined only to one of the domains, which are in a two-way trust relationship with other domains.

Computer Objects and Machine Accounts

By default, Cohesity uses the cluster name as the Machine Account Name and creates the Computer Object on the Active Directory with the same name as the Machine Account Name. You can change the default Machine Account Name while joining the Cohesity cluster and modify it later from the cluster UI.

NOTE: Active Directory expects the “$” character at the end of a Machine Account, but as Cohesity automatically adds the “$” character in the end, there's no requirement to specify the “$” character.

Service Principal Names (SPNs)

When you join a Cohesity cluster or add a new Machine Account to Active Directory, only the SPN for the VIP FQDN is created and added to the newly created Computer Object. Zone Name SPNs are never automatically created and added to the Computer Object, i.e. no automatic SPN creation for Zones for any of the following scenarios:

- Joining the Cohesity cluster to the Active Directory with Zones created
- Creating a new Machine Account in the Cohesity cluster with Zones created
- Creating new Zones in the Cohesity cluster that's already joined to the Active Directory

NOTE: This is by design, as it protects by restricting the exposure of the services only to the Cohesity cluster services. Administrators should enable the services only from the relevant zones by adding the respective SPNs to the Computer Object in the Active Directory.

SPN Handling During Migration

You can associate or register a given SPN with only one Computer Object. So, during data migration, if the hostnames are also getting moved from an existing appliance to the Cohesity cluster, include the below steps in the migration cutover plan:

- Delete/unjoin the old Computer Object with the SPN (which in turn deletes the SPN) or delete the SPN from the old Computer Object.
- Manually create/add the SPN onto the Cohesity cluster's Computer Object (Machine Account).

NOTE: It may take a couple of minutes for the SPN deletion to replicate over all the domain controllers.

For detailed information on the configuration steps, see Appendix C—SPNs & Machine Account (AD Computer Object).

SPNs vs Machine Accounts

There are two ways to serve Active Directory domain services via multiple hostnames from a Cohesity cluster:

1. Add SPNs for the multiple hostnames to single/few Machine Account(s).
2. Join Cohesity cluster multiple times, i.e. one Machine Accounts for each hostname.

NOTE: Cohesity recommends option 2 over option 1.

Group Nesting Strategy

Cohesity recommends and supports the Industry-standard IGUDLA (identities, Global Groups, Universal Groups, Domain Local Groups, and access) group nesting strategy for Active Directory domain group membership and group nesting.

Domain Controllers—Closest, Preferred, and Site Awareness

If the domain controllers are not reachable or there's a high replication delay to the domain controllers, the domain resolution/lookup operations can fail.

Cohesity is currently not Active Directory site-aware. However, this limitation has been mitigated by the implementation of the Domain Controller Monitoring Service, and can be further remediated by using the preferred domain controller.

Domain Controller Monitoring Service is a Cohesity internal service that periodically (in every 20 minutes) checks the status (liveness) of all domain controllers and creates a ranked list of available domain controllers.

- Cohesity fetches the list of domain controllers from the DNS server and creates a ranked list based on their reachability (success, failure, latency, etc.).
- If the preferred domain controllers are specified, then the reachable preferred domain controllers are added to the top of the ranked list.
- Finally, Cohesity leverages Microsoft APIs to locate the domain controller in the closest site, which, if reachable, is added as the topmost entry in the ranked list.

If any of the domain controller in the ranked list becomes unreachable, then they are dropped from the list and are not used for any future domain resolution operations until the Domain Controller Monitoring Service runs again to the background job service to create a new ranked list of domain controllers. If all the domain controllers in the ranked list go unreachable, then the cluster still waits for the next run of the Domain Controller Monitoring Service to create a new ranked list.

Managing Cohesity Cluster via Windows MMC

You can use MMC to perform day-to-day administrative tasks for a Computer Object. MMC offers a common framework in which various snap-ins can run to manage several services from a single interface.

Requirements to connect the Cohesity cluster via MMC:

- The Cohesity cluster's Computer Object (Machine Account) in the Active Directory should have a valid DNS hostname.
- The user logged-in or its group should be added to the cluster's built-in administrators group.

For detailed information on the configuration steps, see Appendix B—DNS IP Address for the Machine Account (AD Computer Object) and Appendix D—Add AD Domain Admins to the Built-in Administrators Group.

Multiple Active Directory Domain Trusts Supportability Matrix

In case of multiple Active Directory domains, follow the below configuration instructions and recommendations based on their Trust types.

No Trust

Table 2: Configuration Instructions—No Trust

TRUST TYPE	CONFIGURATION INSTRUCTIONS/COMMENTS IN THE COHESITY CLUSTER
No Trust	All domains act as standalone domains and so the “Trusted Domain discovery” should stay DISABLED.

Two-way Transitive Trust

Table 3: Configuration Instructions—Two-way Transitive Trust

TRUST TYPE	CONFIGURATION INSTRUCTIONS/COMMENTS IN THE COHESITY CLUSTER

Parent-Child
Tree-Root
Forest
Shortcut
	• The Cohesity cluster should be joined only to one of the domains.
	• The chosen domain must be able to resolve all the users, groups, and objects connecting to the Cohesity cluster.
	• For the best response time, it’s always recommended to choose the domain with the maximum number of users, groups, and objects connecting to the Cohesity clusters.
	• “Trusted Domain discovery” must be ENABLED and ensure that:
	  ○ Domains with two-way transitive trust are not added to the “Excluded” list.
	  ○ All domains with one-way transitive trust are added to the “Excluded” list.
	  ○ Any misbehaving domains are added to the “Excluded” list.

One-way Transitive Trust

Table 4: Configuration Instructions—One-way Transitive Trust

TRUST TYPE	CONFIGURATION INSTRUCTIONS/COMMENTS IN THE COHESITY CLUSTER

Forest	• The Cohesity cluster should be joined to all of the domains individually as different Computer Objects (via Machine Accounts).
	• if there are no further domains with two-way transitive trust, ensure that the “Trusted Domain discovery” is DISABLED.
	• If there are further domains with two-way transitive trust, ensure that “Trusted Domain discovery” is ENABLED, and:
	  ○ Domains with two-way transitive trust are not added to the “Excluded” list.
	  ○ All domains with one-way transitive trust are added to the “Excluded” list.
	  ○ Any misbehaving domains are added to the “Excluded” list.
	Note: Cross-domain group membership lookups and resolution do not work, as domains are considered independent.

Shortcut	

Non-transitive Trust

Table 5: Configuration Instructions—Non-transitive Trust

TRUST TYPE	CONFIGURATION INSTRUCTIONS/COMMENTS IN THE COHESITY CLUSTER
External	• The Cohesity cluster should be joined to all of the domains individually as different Computer Objects (via Machine Accounts). • If there are further domains with one-way/two-way transitive trust, then the recommendations in one-way/two-way transitive trust must be followed.
Realm	As there are many third-party realm providers and their configurations vary based on the realm software vendor/provider, not all realm-based transitive/non-transitive trusts may be supported out of the box. Contact Cohesity Support with your configuration details to provide the necessary instructions.

Advanced Configuration

Cohesity offers advanced settings that are not part of the general administration options. The reasons for not exposing these advanced settings range from their sparse usefulness to the adverse impact they can cause if configured improperly. Cohesity restricts the documentation for these advanced settings to Cohesity Internal Personnel only.

Advanced Configuration from SmartFiles KB article includes a list of commonly used advanced configuration settings. Please contact Cohesity Support to get them configured on your Cohesity cluster.

Table 6: Advanced Configuration Settings

S NO.	ADVANCED FEATURE/SETTING
1	When a client authenticates using NTLM while accessing a share using SMB protocol, the user's group membership is also retrieved from AD along with the authentication. This method returns complete group membership in most of the common configurations. However, it does not return full group membership in certain configurations—especially nested groups across multiple domains. In such a case, we can enable group membership to be retrieved from AD using the LDAP protocol.
2	By default, NTLMv1 is enabled on Cohesity. However, you can disable NTLMv1 to improve security.
3	Support for nested group membership in the IGUDLA model (described in the Nested Group Strategy section) is disabled by default, as searching for nested group membership is usually resource intensive. However, you can enable it, if required.
4	The discovery of trusted domains and user's group membership could be expensive in a complex AD tree or forest. To limit domain discovery or group search, Cohesity searches for trust domains only up to 3 levels deep (starting from the primary domain; the primary domain is considered as level = 0). However, this value might need to be increased in certain AD configurations.
5	There are three types of AD groups viz. Domain Local groups (DLG), Universal Groups (UG), and Global Groups (GG). Out of these three, DLG groups can have principals from trusted domains in the same forest as well as trusted domains from a different forest. When this flag is enabled, we do cross-domain group lookups from within the forest as well as from different forests.

Appendix A—Brief Overview of Active Directory

Active Directory is a directory service developed by Microsoft for Windows domain networks. A server running the AD DS role is called a domain controller. It authenticates and authorizes all users and machines in the Active Directory domain. Active Directory uses LDAP versions 2 and 3, and Microsoft's version of Kerberos.

Active Directory Authentication (Kerberos vs NTLM)

Microsoft Active Directory provides two protocols for authentication:

Table 7: Active Directory Authentication Protocols

KERBEROS	NTLM
Kerberos is the default authentication protocol for Active Directory (Windows 2000 onwards).	NTLM was the default authentication protocol in the older versions of Windows operating system (prior to Windows 2000).
Kerberos uses a two-part process that leverages a Ticket Granting Service (TGS) or Key Distribution Center (KDC).	NTLM, originally developed by Microsoft, relies on a three-way handshake between the client and server to authenticate a user.
	NTLM has two versions: NTLMv1 and NTLMv2. NTLMv2 is more secure than NTLMv1.

Even though the Kerberos protocol is Microsoft's default authentication protocol, NTLM is still maintained in all Windows systems. If Kerberos fails to authenticate the user, the system will attempt to use NTLM. NTLM remains widely deployed even on new systems in order to maintain compatibility with legacy clients and servers.

Time Synchronization

Microsoft Active Directory uses the Kerberos v5 protocol, which is based upon the time-stamped tickets. So, all the domain controllers and computers must be set to the same time and date for kerberization (authentication) to work seamlessly.

You can set maximum tolerance for differences in the computer clock between a client and the domain controller. The difference in the computer clock must always be less than this defined maximum time, for the timestamps to be considered while authenticating the session. By default, the maximum tolerance for computer clock synchronization is set to a value of 5 minutes.

Computer Object and its Attributes—Machine Accounts, DNS Name, and SPNs

To join a computer to an Active Directory, a computer uses a Machine Account as the login name and creates a Computer Object with the same name as the Machine Account. While the Machine Account is used to run all the services, the Computer Object holds all the attributes for the computer in the Active Directory (For example: DNS hostname, Managed by, Members of, OU, SPNs, etc). This Computer Object gets created either on the default OU (Organization Unit) path or the OU path specified while joining the computer.

- The Machine Account always has a “$” character at the end. The Computer Object is created with the same name as the Machine Account, without the final "$" character in the end.
- While joining the Computer Object, if DNS hostname is not specified, then the value of the DNS hostname on the Computer Object by default is set to **computerName.fullDomainDnsName**; where— **computerName** is the Computer Object, and **fullDomainDnsName** is the DNS name of the domain.

SPN is a unique identifier of a service instance. SPNs are used by Kerberos authentication to associate a service instance with a service logon account. When a client wants to connect to a service, it locates an instance of the service, composes an SPN for that instance, connects to the service, and presents the SPN for the service to authenticate. The SPN must be registered on the Computer Object that the service instance uses to log on.

A given SPN can be registered on only one Computer Object. But a given service instance can have multiple SPNs if there are multiple names that clients might use for authentication. An SPN always includes the name of the host computer on which the service instance is running, so a service instance might register an SPN for each name or alias of its host. For more details, see Microsoft’s documentation on Service Principal Names.

Disjoint Namespace

A disjoint namespace occurs when one or more domain member computers have a primary DNS suffix that does not match the DNS name of the Active Directory domain of which the computers are members.

A disjoint namespace should work in the following situations:

When a forest with multiple Active Directory domains uses a single DNS namespace (DNS Zone):

AD Domains	amer.foobar.com; emea.foobar.com; apjc.foobar.com
DNS Zone	foobar.com

When a single Active Directory domain is split into separate DNS namespaces:

AD Domain	foobar.com
DNS Zones	amer.foobar.com; emea.foobar.com; apjc.foobar.com

Controller Proximity and Site Awareness

In an enterprise environment, to provide load-balancing and fault-tolerance to an Active Directory, more than one domain controller is usually configured to serve the Active Directory domain. These domain controllers are always in replication with each other. AD DS uses a multimaster, store-and-forward method of replication. A domain controller communicates directory changes to a second domain controller, which then communicates to a third, and so on, until all domain controllers have received the change.

These domain controllers are usually in different subnets, VLANs, sites, or even geographies. To achieve the best balance between reducing replication latency and reducing traffic, site topology controls Active Directory replication by distinguishing between replication that occurs within a site and replication that occurs between sites. The replication interval between sites can vary based on the site link replication interval set by the Active Directory administrator.

Within sites, replication is optimized for speed, data updates trigger replication, and the data is sent without the overhead required by data compression. Conversely, replication between sites is compressed to minimize the cost of transmission over WAN links. When replication occurs between sites, a single domain controller per domain at each site collects and stores the directory changes and communicates them at a scheduled time to a domain controller in another site.

For more details, see Microsoft's documentation on Site Link Properties.

Group Scope and Group Nesting

Groups in Microsoft environment and Active Directory are characterized by a scope that identifies the extent to which the group is applied in the domain tree or forest. The scope of the group defines where the group can be granted permissions.

Table 8: Group Scope and Group Nesting

GROUP SCOPE	MEMBERS FROM THE SAME DOMAIN	MEMBERS FROM TRUSTED DOMAIN IN THE SAME FOREST	MEMBERS FROM TRUSTED DOMAIN DIFFERENT FOREST
Local	• Users • Computers • Global Groups • Domain Local Groups • Universal Groups • Local Users defined on the same computer as the local group	• Users • Computers • Global Groups • Universal Groups	• Users • Computers • Global Groups
Domain Local	• Users • Computers • Global Groups • Domain Local Groups • Universal Groups	• Users • Computers • Global Groups • Universal Groups	• Users • Computers • Global Groups
Universal	• Users • Computers • Global Groups • Universal Groups	• Users • Computers • Global Groups • Universal Groups	NA
Global	• Users • Global Groups	NA	NA

Group Nesting is the process of adding one group to another group. This will allow you to help better manage and administer your environment based on business roles, functions, and management rules.

Active Directory Trust Relationships

Multiple Active Directory domains, when not in a trust relationship, act as standalone domains. Trust relationships can be established between the domains for secure communication between users, groups, and objects in the individual domains. Trusts act as bridges that only allow validated authentication requests to travel between domains. How a trust passes authentication requests depends on how it's configured. Trusts can be one-way or two-way, and can be transitive or non-transitive.

The flow of communication over trusts is determined by the direction of the trust. Trusts can be configured in one of the following ways:

- One-way: Provides access from the trusted domain to resources in the trusting domain.
- Two-way: Provides access from each domain to resources in the other domain.

Trusts are also configured to handle additional trust relationships in one of the following ways:

- Non-transitive: The trust exists only between the two trust partner domains.
- Transitive: Trust automatically extends to any other domains that either of the partners trusts.

Below are the various types of Trust that can be configured in Microsoft Active Directory:

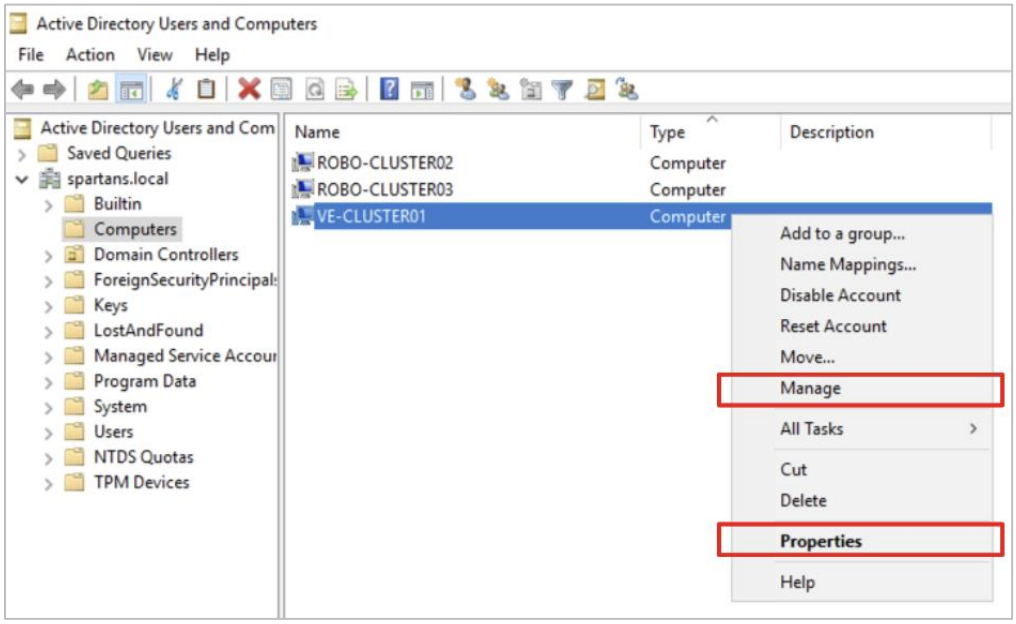
Table 9: Active Directory Trust Types

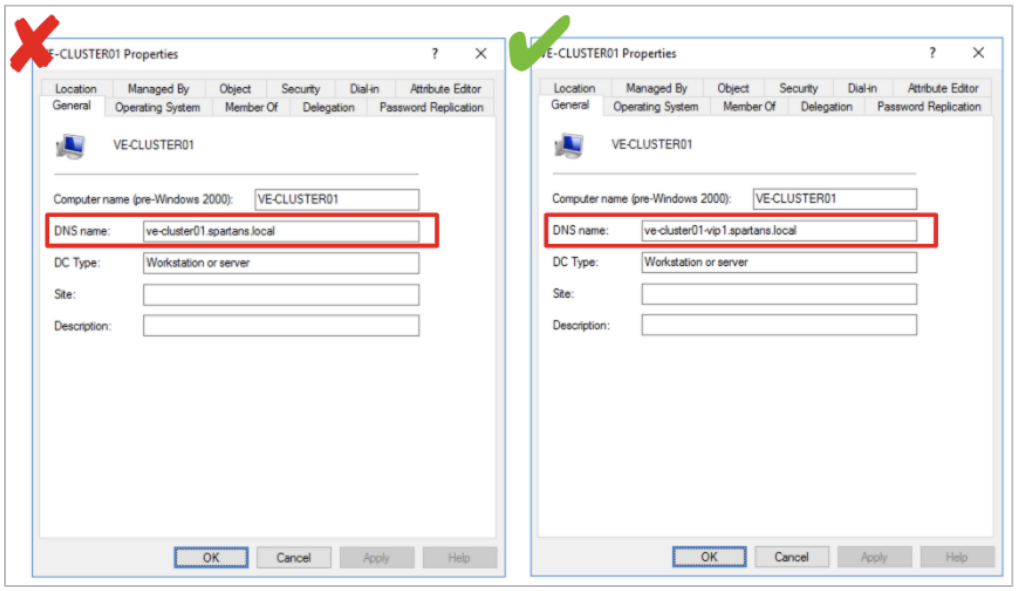
TRUST TYPE	TRANSITIVITY	DIRECTION	DESCRIPTION
Parent-Child	Transitive	Two-way	• Created when a new subdomain (child) is added to the root domain (parent). Authentication requests made from the child domain(s) flow upward through their parent to the trusting domain. • Created automatically when a child domain is added.
Tree-Root	Transitive	Two-way	• Created when a new domain tree is added to an existing AD forest. • Created automatically when a new tree is added to the AD forest.
Forest	Transitive	One-way or Two-way	• Used when access to resources between forests is needed. • Created manually.

TRUST TYPE	TRANSITIVITY	DIRECTION	DESCRIPTION
Shortcut	Transitive	One-way or Two-way	• Used when authentication speed is needed between domains in a forest(s) that might be separated by several domains or domain trees. • Create manually.
Realm	Non-transitive or Transitive	One-way or Two-way	• Used when access to resources is needed between a third-party directory service(s) and Active Directory (i.e. Windows Kerberos V5 realm). • Created manually.
External	Non-transitive	One-way or Two-way	• Used when access to resources in a separate forest is needed and forest trust is not set up. • Created manually.

Appendix B—DNS IP Address for the Machine Account (AD Computer Object)

Check the DNS hostname in the Computer object for Cohesity.





Modify the DNS hostname in the Computer object for the Cohesity cluster via either of the following methods:

1. Modify the DNS hostname from Cohesity cluster from:

   Settings → Access Management → Active Directory → *{Select The Domain}* → Edit:*{Machine Accounts}* → Edit:*{Machine Account}* → DNS hostname (optional)

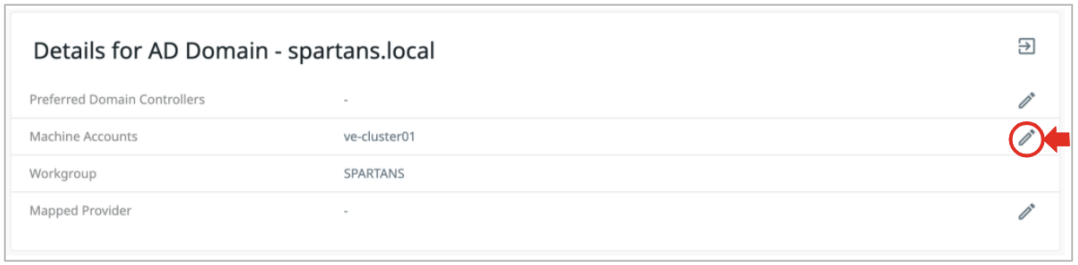
   > **NOTE**: While modifying an existing Machine Account, always enable the setting—"**Use the above Machine Accounts even if they already exist in AD Domain**."

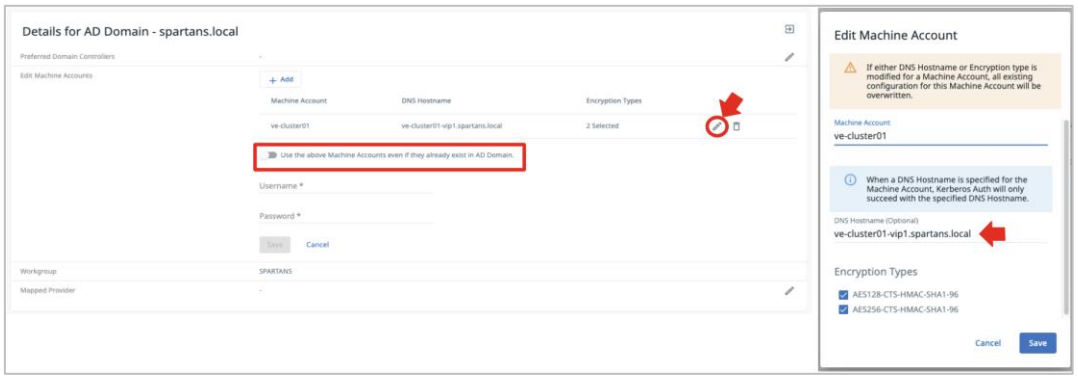
2. Modify the **DNShostname** attribute on the **Computer Object** with ADSI Edit.

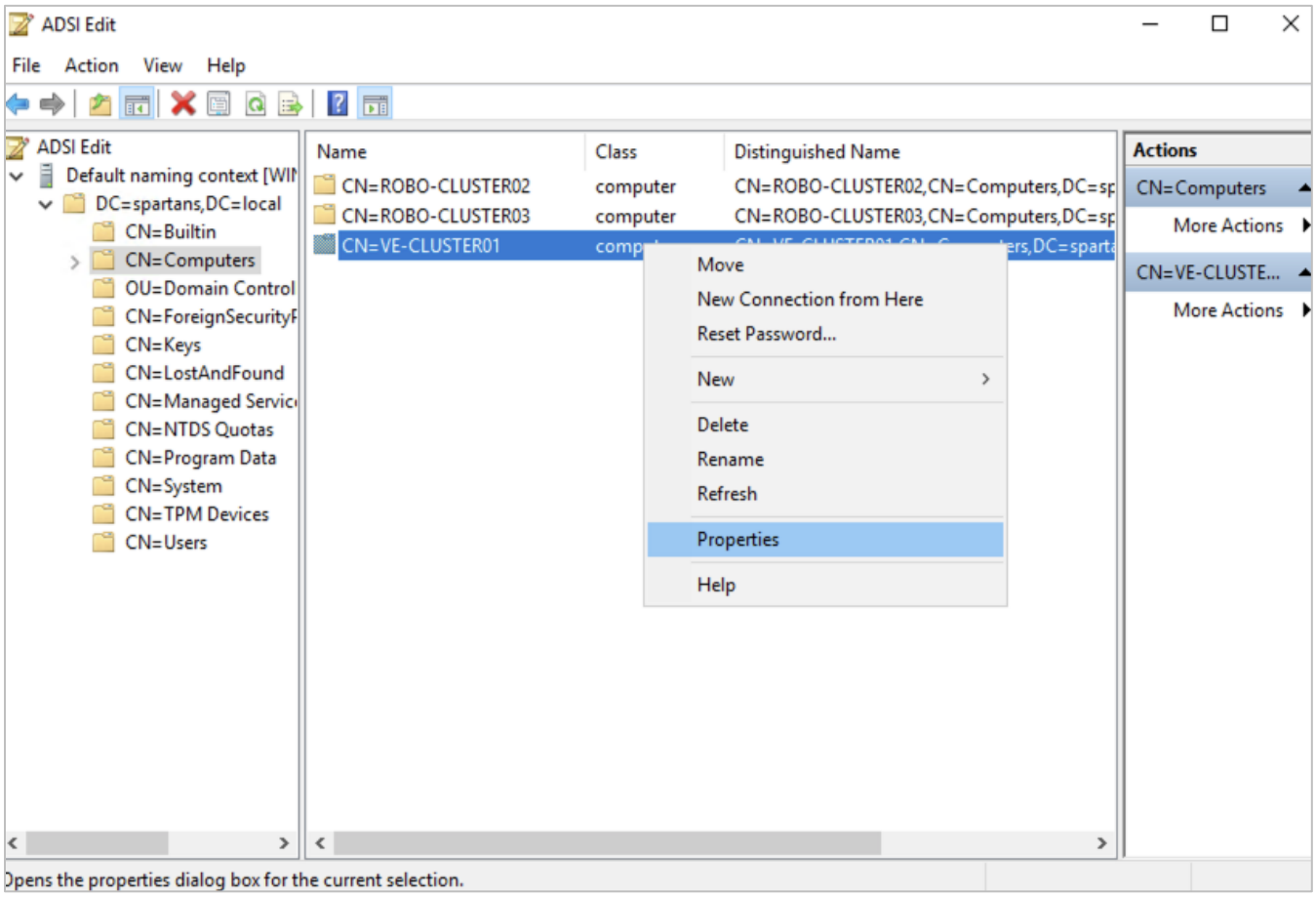
   ADSI Edit (Active Directory Services Interfaces Editor) is a low-level editor for Active Directory Domain Services/Active Directory Lightweight Directory Services. It allows you to view, modify, create, and delete any object in Microsoft's AD DS/LDS. ADSI Edit is part of Active Directory Tools provided by Microsoft. Please check with your Active Directory administrator for more information.

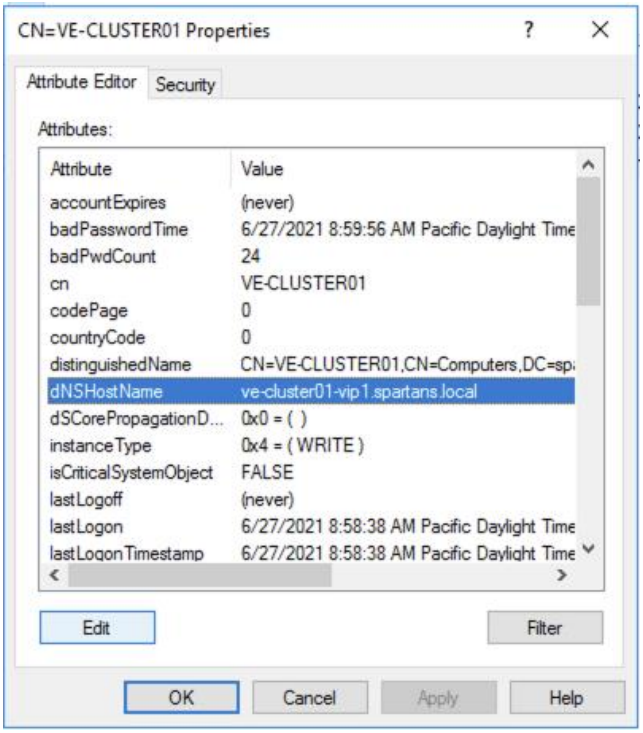
   Here is how you modify the hostname of a SmartFiles cluster with the following properties:

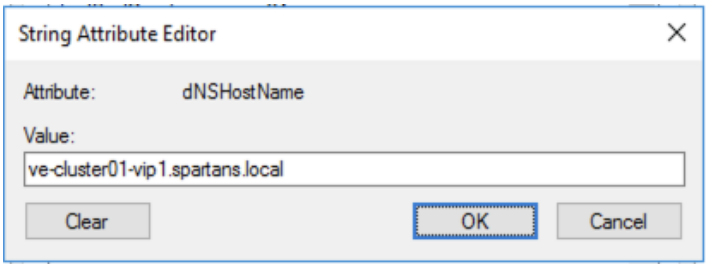
Cluster Name	VE-Cluster01
VIP FQDN	ve-cluster01-vip.spartans.local
DNS Domain	spartans.local
AD Domain	spartans.local
Machine Account (Computer Object)	ve-cluster01
Zone Name	apps.spartans.local, archives.spartans.local






ADSI Edit
File Action View Help
ADSI Edit
Default naming context [WIN
DC=spartans,DC=local
CN=Builtin
CN=Computers
OU=Domain Control
CN=ForeignSecurityP
CN=Keys
CN=LostAndFound
CN=Managed Servic
CN=NTDS Quotas
CN=Program Data
CN=System
CN=TPM Devices
CN=Users
Name
Class
Distinguished Name
CN=ROBO-CLUSTER02
computer
CN=ROBO-CLUSTER02,CN=Computers,DC=sp
CN=ROBO-CLUSTER03
computer
CN=ROBO-CLUSTER03,CN=Computers,DC=sp
CN=VE-CLUSTER01
Move
New Connection from Here
Reset Password...
New
Delete
Rename
Refresh
Properties
Help
Actions
CN=Computers
More Actions
CN=VE-CLUSTE...
More Actions
Opens the properties dialog box for the current selection.


CN=VE-CLUSTER01 Properties
Attribute Editor
Security
Attributes:
Attribute
Value
accountExpires
(never)
badPasswordTime
6/27/2021 8:59:56 AM Pacific Daylight Time
badPwdCount
24
cn
VE-CLUSTER01
codePage
0
countryCode
0
distinguishedName
CN=VE-CLUSTER01,CN=Computers,DC=sp
dNSHostName
ve-cluster01-vip1.spartans.local
dSCorePropagationD...
0x0 = ( )
instanceType
0x4 = (WRITE)
isCriticalSystemObject
FALSE
lastLogoff
(never)
lastLogon
6/27/2021 8:58:38 AM Pacific Daylight Time
lastLogonTimestamp
6/27/2021 8:58:38 AM Pacific Daylight Time
Edit
Filter
OK
Cancel
Apply
Help


String Attribute Editor
Attribute:
dNSHostName
Value:
ve-cluster01-vip1.spartans.local
Clear
OK
Cancel

Appendix C—SPNs & Machine Account (AD Computer Object)

SPNs for the Zone Names can be created via either of the following methods:

1. Run the `setspn` command.

   `Setspn` is a command-line tool that is built into Windows Server. It is available if you have the AD DS server role installed. To use `setspn`, you must run the `setspn` command from an elevated command prompt.

   To open an elevated command prompt, click Start, right-click Command Prompt, and then click Run as Administrator.

   For example, for the SmartFiles Cohesity cluster with:

Cluster Name	VE-Cluster01
VIP FQDN	ve-cluster01-vip.spartans.local
DNS Domain	spartans.local
AD Domain	spartans.local
Machine Account (Computer Object)	ve-cluster01
Zone Name	apps.spartans.local, archives.spartans.local

- List all the currently registered SPNs for the Computer Object VE-Cluster01:

  ```
  setspn -l ve-cluster01
  ```

- Reset to default registered SPNs for the Computer Object VE-Cluster01:

  ```
  setspn -r ve-cluster01
  ```

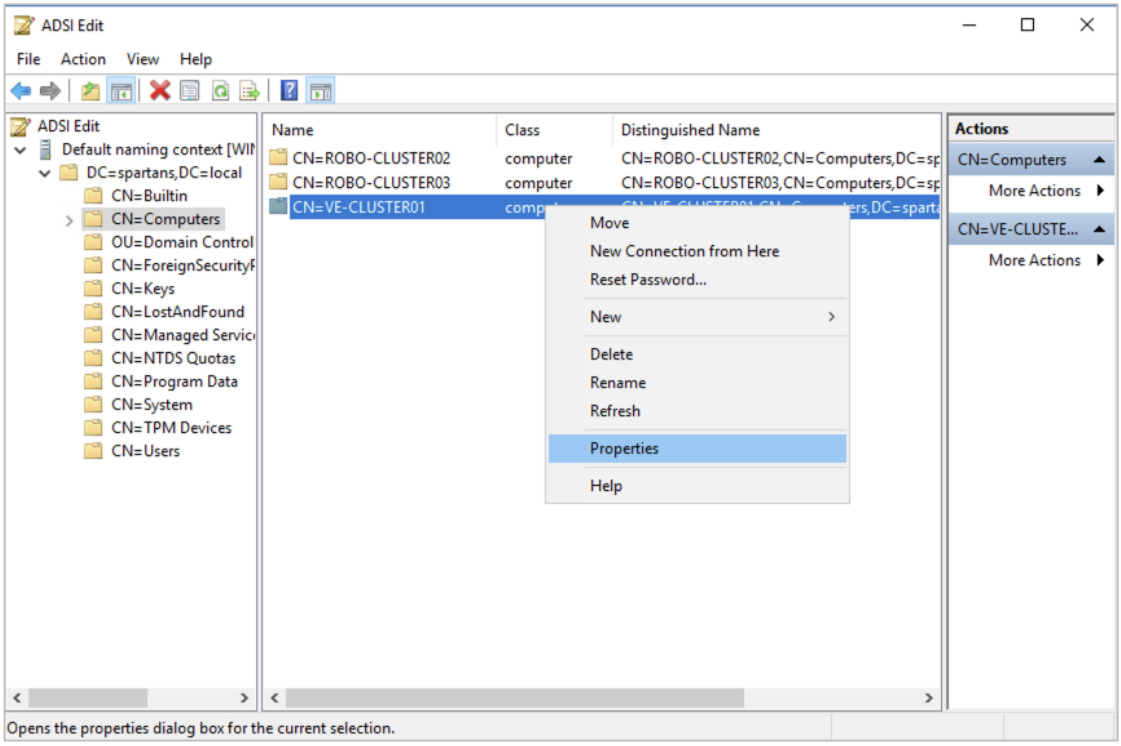
- Add a new SPN HOST/apps.spartans.local to the Computer Object VE-Cluster01:

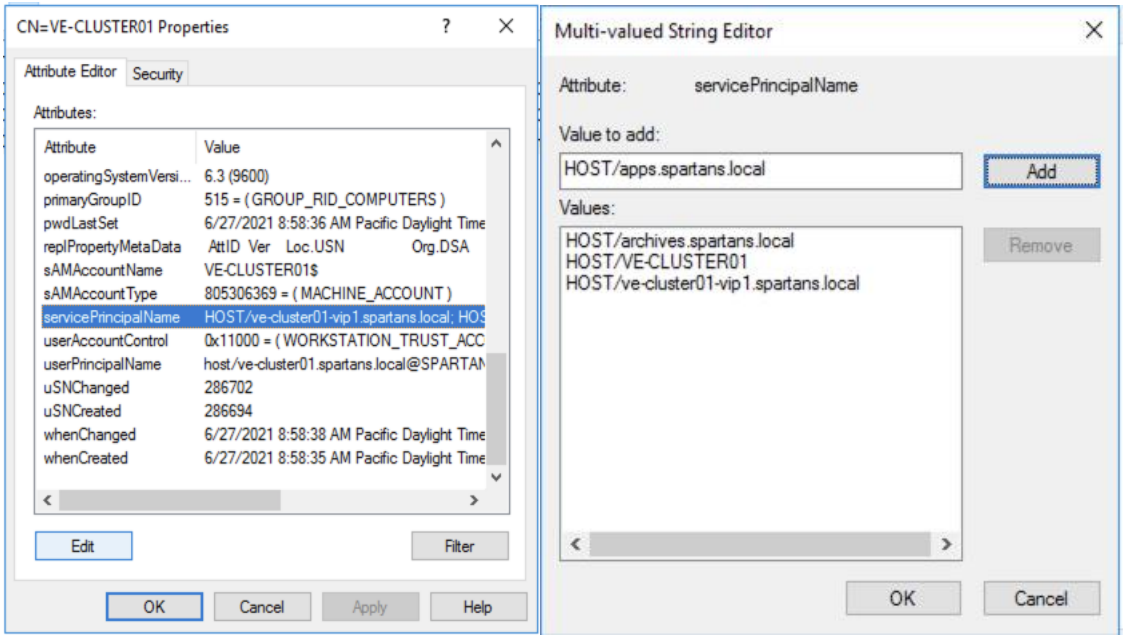
  ```
  setspn -s HOST/apps.spartans.local ve-cluster01
  ```

- Remove SPN HOST/apps.spartans.local to the Computer Object VE-Cluster01:

  ```
  setspn -d HOST/apps.spartans.local ve-cluster01
  ```

2. Modify servicePrincipalName attribute on the **Computer Object** via with ADSI Edit.



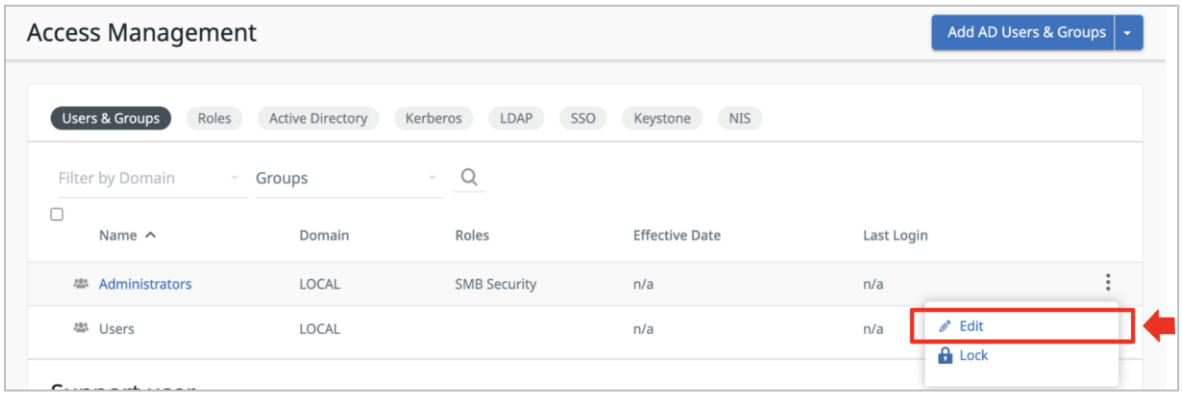


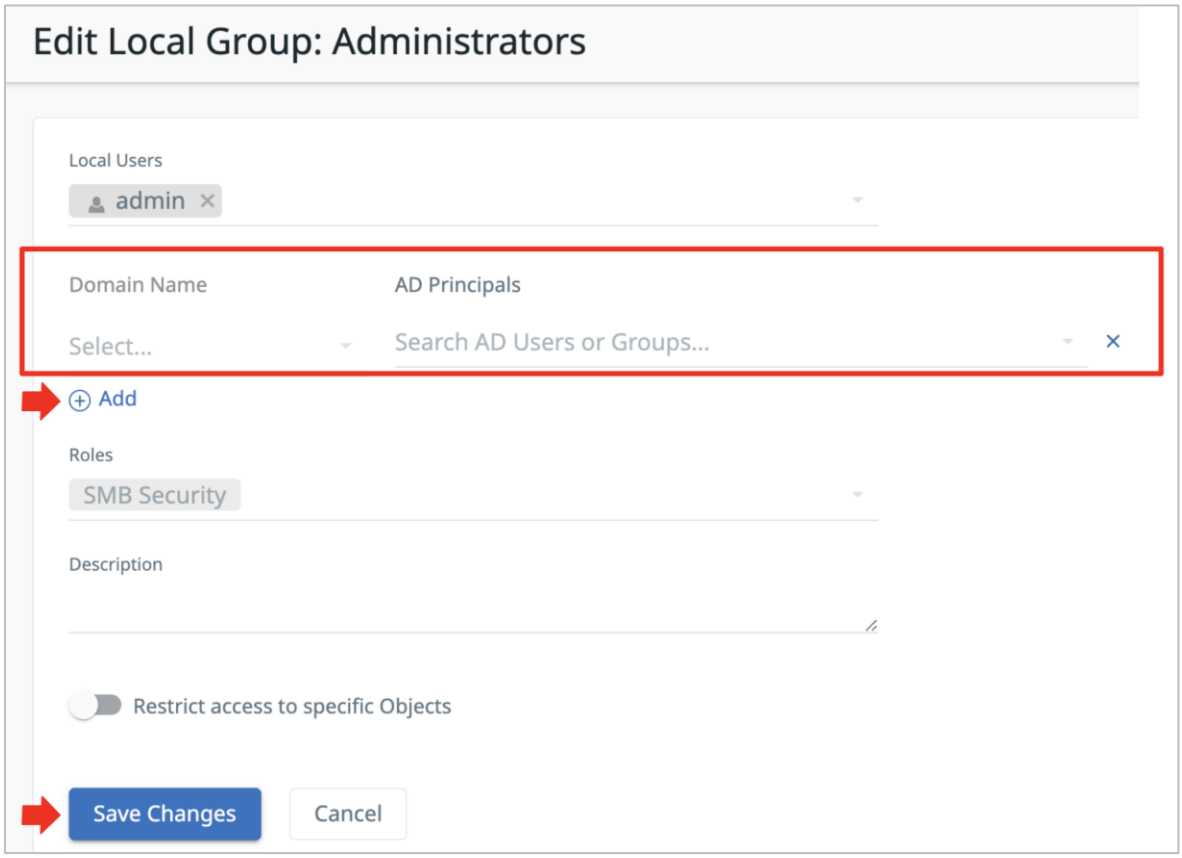
Appendix D—Add AD Domain Admins to the Built-in Administrators Group

You can add Domain Admin users and groups from Active Directory to the built-in administrators group of the Cohesity cluster via either of the following methods:

1. Add AD Domain Admins to the built-in administrators group from the Cohesity cluster:

   Settings → Access Management → Users & Groups → *Administrators* → Add → *{Domain Name}* → *{AD Principals}*





2. Add AD Domain Admins to the built-in administrators group from the MMC.

   There are two ways to access MMC:

   a) Open **Active Directory Users & Computers**, search for the Computer Object, and then right-click and select Manage.

      Requirements for this method to work:

      - Cohesity cluster's Computer Object (Machine Account) in the Active Directory should have a valid DNS hostname.
      - The user logged-in or its group should be added to the cluster's built-in administrators group.

   b) Open **Computer Management,** and then go to **Actions → Connect to another computer** and specify DNS hostname for the Cohesity cluster. This DNS hostname can either be a VIP hostname or Zone Name.

      Requirements for this method to work:

      - The user logged-in or its group should be added to the cluster's built-in administrators group.

Technical Support and Resources

Cohesity Support Portal provides you access to a robust, on-demand, and detailed knowledge base, along with high-quality services to boost your experiences with Cohesity products.

Cohesity Product Documentation provides you access to the latest product documentation to support your deployment of Cohesity products including technical guides and third-party software support matrix for Cohesity Data Protection.

Cohesity Developer Portal provides you ready-to-use integrations with the automation and orchestration tools of your choice to streamline operations.

Related Resources

KBs / WHITE PAPERS / BLOGS
White Paper: Cohesity: Identity and Access Management for File Services This white paper describes the identity and access management for file services on Cohesity 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 multiprotocol environment.
White Paper: SmartFiles Internal Load Balancer Configuration Guidelines SmartFiles Internal Load Balancer enables Cohesity clusters to distribute the client connections across multiple nodes. This removes the dependency from External DNS and its cumbersome configuration.
Knowledge Base: Recommended settings when using Cohesity SmartFiles (NFS, SMB, and S3) Some generic recommendations and best practices for configuring and optimizing a Cohesity cluster for SmartFiles (NFS, SMB, and S3).
Knowledge Base: Advanced Configuration from SmartFiles KB article Advanced settings that are not part of the general administration options. Cohesity restricts the documentation for these advanced settings to Cohesity Internal Personnel only. Please contact Cohesity Support to get them configured on your Cohesity cluster.

Your Feedback

Was this document helpful? Send us your feedback!

About the Authors

Akshay Kumar is a Senior Product Solutions Engineer at Cohesity. In his role, he focuses on SmartFiles File and Object services.

Other essential contributors include:

- Adaikkappan Arumugham, Technical Marketing Engineering
- Ruby Garg, Technical Marketing Engineering
- Balakumaran Govindhasamy Radhakrishnan, Engineering
- Pralay Dakua, Engineering
- Subash Babu, Technology Editor-Technical Marketing

Document Version History

VERSION	DATE	DOCUMENT HISTORY
1.0	Dec 2021	First release

ABOUT COHESITY

Cohesity radically simplifies data management. We make it easy to protect, manage, and derive value from data -- across the data center, edge, and cloud. We offer a full suite of services consolidated on one multicloud data platform: backup and recovery, disaster recovery, file and object services, dev/test, and data compliance, security, and analytics -- reducing complexity and eliminating mass data fragmentation. Cohesity can be delivered as a service, self-managed, or provided by a Cohesity-powered partner.

Visit our website and blog, follow us on Twitter and LinkedIn and like us on Facebook.