COHESITY

Version 3.0

August 2024

Cohesity Physical Agent Best Practice and Deployment Guide

Make Cohesity Physical Agent Operations Simple and Scalable

ABSTRACT

Cohesity DataProtect supports Windows, Linux, AIX, HP-UX, and Solaris Physical Agents and clustering solutions, which allow users to back up their data on the Cohesity Data Cloud platform. This solution brings a snapshot-based, incremental-forever backup approach to a scalable and robust platform for your physical servers. This guide helps you implement Physical Agents on supported operating systems and explains the best practices.

Table of Contents

Tables

Introduction to Physical Agents

Cohesity provides a physical backup agent for data protection that can be installed on customer servers. Cohesity Physical Agents are binaries that can be installed on supported operating systems to protect data. After installing the Cohesity Agent on a server, the Cohesity cluster can use a block-based or file-based Protection Group for backup and recovery. The Cohesity cluster leverages agents to interface with file systems, applications, and databases to facilitate the protection of data on production systems.

Cohesity Data Protection–Supported Physical Agents and Deployment Methodology

The following section discusses the prerequisites and the planning required for Physical resource backup. To have a healthy copy of the production system backup, we must plan the backup configuration wisely to avoid interruption.

Planning and Preparation

Before using a Physical Agent for data protection, ensure that all the following prerequisites are met and firewall ports are open. For port information, refer to the Documentation.

Cohesity Version 6.8.1 and higher support the following operating systems:

- Windows
- Linux
- AIX
- Solaris
- HP-UX

Table 1: Cohesity Data Protection-Supported Physical Agents and Deployment Methodology

Physical Agents	Supported Versions
Windows	Core Server 2022,2019, 2016 Servers 2022,2019,2016, 2012, 2012R2, 2008 R2, Windows 10 - Desktop
RHEL	9.0-9.2, 8.0-8.8, 7.0-7.9, 6.7+, 5.8-5.11 Rocky Linux 8.7,8.8,9.0,9.1 X64 and PowerPC little-endian (SLES 12 SP5, 15SP3)
Debian	11.x, 10.0, 9.6
SUSE	15.3, 15.0,12.3, 12 SP4,11SP4 Open SUSE 15.1
Centos	5.10, 6.0+, 7.0-7.9, 8.0,8.3
AIX	7.3, 7.2, 7.1, 6.1 (IBM POWER7, POWER8, POWER9)

Physical Agents	Supported Versions
Solaris	11, 10
Oracle Enterprise Linux	5.8, 6.x, 7.0-7.9, 8.2-8.5
Ubuntu	14.x, 16.x, 18.x, 19.x, 20.x, 22.04
HP-UX	11i v3 (B.11.31)

For more information, see the list of supported software for physical servers.

Hardware and Software Requirements

To install Cohesity Physical Agent, you must meet the following hardware and software requirements.

Table 2: Minimum Requirements

Operating System	Minimum CPU	Minimum Memory	Minimum Supported Operating System	Free Disk Space	Software Requirement
RHEL	1 core	300 MB	RHEL 6	400 MB	nfs-utils, rsync, lsof, wget
Debian	1 core	300 MB	Ubuntu 14	400 MB	nfs-utils, rsync, lsof, wget
SLES	1 core	300 MB	SLES 11 SP4	400 MB	nfs-utils, rsync, lsof, wget, libcap-progs
OEL	1 core	300 MB	OEL 6	400 MB	nfs-utils, rsync, lsof, wget
Solaris	0.5 core	1.5 GB	Solaris 10	100 MB	Java 1.8, sudo
AIX	0.25 core	1 GB	AIX 6.1	100 MB	Java 1.8, sudo
HP-UX	1 core	1 GB	HP-UX 11.31	100 MB	Java 1.8, sudo

Minimum Permission and Dependency

The user must fulfill the minimum permission criteria for installing Cohesity Physical Agent on the server. Ensure you have minimum permissions to install, register, back up, and restore and that the physical server has a pre-installed dependent package. The requirement varies according to the operating system for every physical source.

For details of the minimum permission required for each OS, package dependency, and steps for installation and management, see the Cohesity Documentation for the operating system you use.

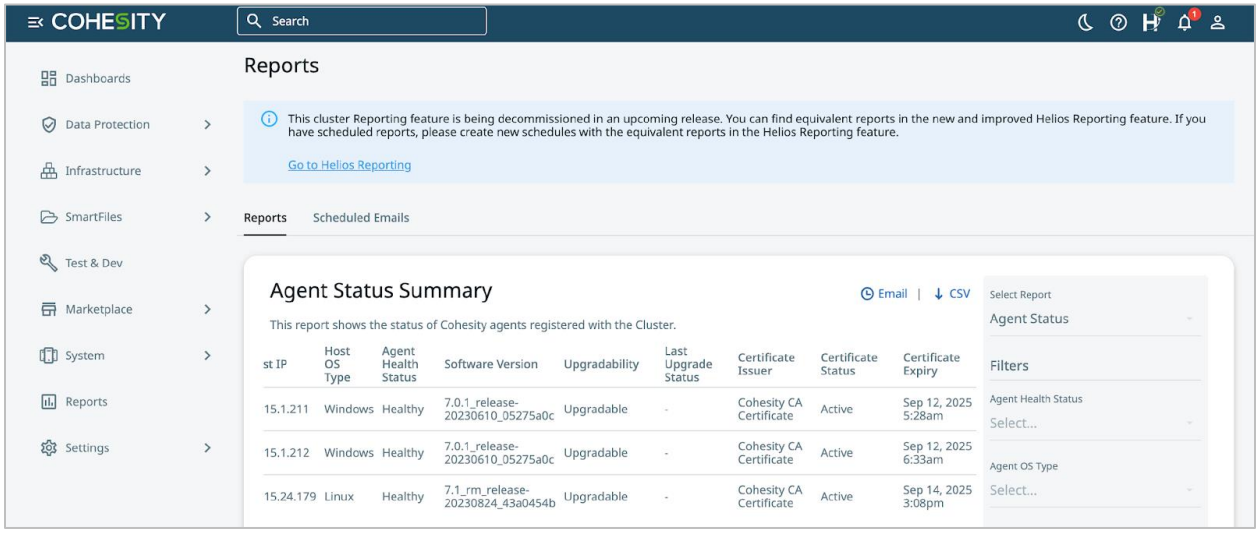
Cohesity CA Certificate - Public Key Infrastructure (PKI)

From version 6.8.1_u5 and 7.1 (in the 7.x family), Cohesity introduced the Certificate Authority (CA) certificate service to enhance security, which will sign certificates used by agents. An intrinsic automation feature will also provision and automatically upgrade prior certificates without user intervention.

You can cross-verify the certificate type of Cohesity CA as "kCohesityCert" in the entity hierarchy. The output should be as follows:

```
agent_cert_info {
      expiry_time_epoch: 2002006800000000
      cert_type: kCohesityCert
    }
```

Cohesity provides the option to Verify from the Agent Status reports, and you can see the certificate issuer.



Cohesity cluster keeps track of the certificates' expiry and creates alerts in the Cohesity UI.

The following types of alerts would be generated -

- An information alert is provided daily when the certificate expires in about 180 days.
- A daily warning alert when the certificate expires in less than 60 days.
- A critical alert when the certificate expiry time is less than 30 days.

Re-register Cohesity Agent

To back up a physical server, the second step after installation of the agent is to register it as a source with the Cohesity Cluster. To register the physical server with the Cohesity Cluster, follow the documentation.

Before Cohesity Cluster Version 7.0, if you reinstall the agent on the physical server, you must register it again with the Cohesity Cluster, resulting in a full backup in the next run. From 7.0, Cohesity introduced a new feature, “Re-register Cohesity Agent.” This feature overcomes multiple challenges, such as -

- Incremental backups can run after this procedure
- No manual information is required for registration.

This feature allows you to save the agent setting on the cluster, perform an agent registration with the same settings, and allow the existing file-based protection job to continue to work with the new agent without any intervention. However, block-based protection may need to be modified if specific volumes under the job have changed before re-registration.

NOTE:

- The first backup after re-registration for block-based protection jobs will be a full backup.
- Re-register functionality is currently supported for Windows and Linux servers.
- Re-registration of the agent will reinstate the agent flags, which are altered on the agent side. You can leverage the REST API to re-register the agent.

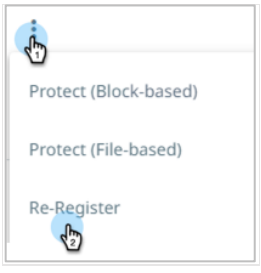
To re-register the agent, follow the below process:

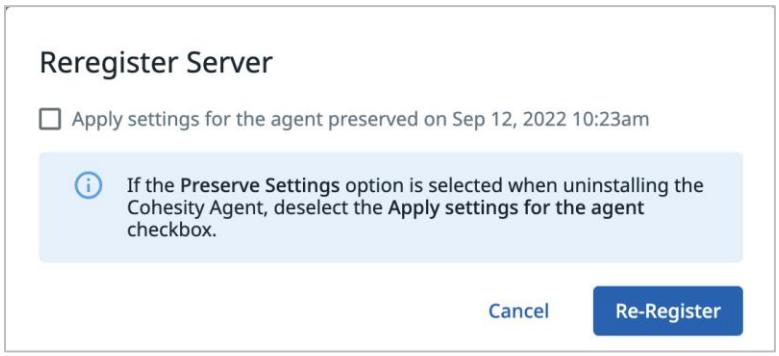
1. Select **Data Protection** > **Sources**.
2. In the source hierarchy, navigate to the Windows or Linux server you want to re-register, mouse over its actions menu, and click **Re-Register**.

   Click to apply the settings for the agent preserved on the **<date and time>** checkbox to restore the server settings.

   If this checkbox is not selected, then the settings saved on the cluster are not applied to the newly registered agent.

   Note: During agent uninstallation of a Cohesity Windows or Linux agent, there is an option to preserve settings in the Windows registry or config file for Linux. If you had selected the Preserve Settings option during the uninstall process then deselect the above option during the re-registration workflow.





3. Click **Re-Register**.

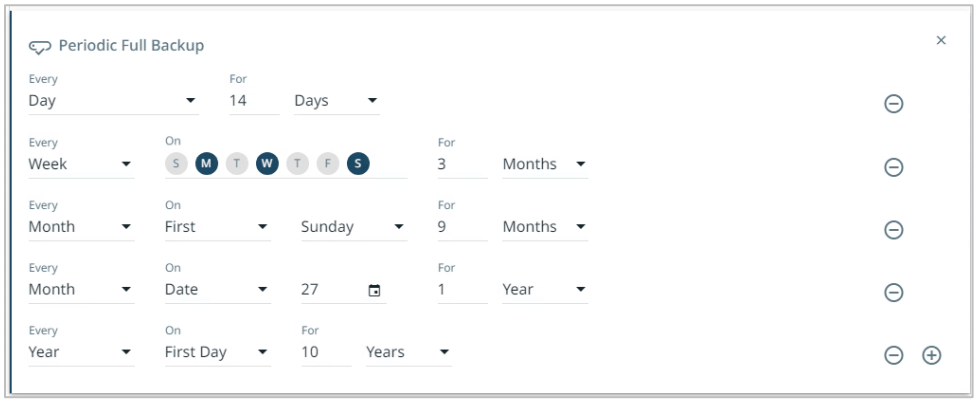
Calendar Based Scheduling

Frequency-based or interval-based scheduling is one of the most popular forms of scheduling our customers use. It is the primary choice for running daily/weekly backups. However, customers need the functionality to set a backup schedule based on a fixed day/date without creating another protection group or policy to reduce administrative overhead. The most popular use cases are governance and auditing needs for long-term retention or compliance.

From version 7.0, Cohesity introduced the feature "Calendar-based scheduling." With a custom retention set, this feature allows you to schedule a full backup on the Policy level for Daily, Weekly, Monthly, and Yearly recurrences. This feature enables granular control for scheduling, as a user can select specific days and dates. There is a 60-minute window that can overlap with another schedule. The scheduling options area -

- **Day Scheduling Policy:** This will allow you to run the full backup daily.
- **Weekly Scheduling Policy:** This policy allows users to run the full backup on specific days of the week, for example, Monday or Wednesday.
- **Monthly Scheduling Policy:** This policy allows the user to select a specific day of the month, for example, the First day or the Last day of the month, with the day's option, and the user can select the specific date to run the backup.
- **Yearly Scheduling Policy:** This policy allows users to select the first or last day of the year.

Retention: The user can apply the specific retention with the calendar-based scheduling mentioned below.

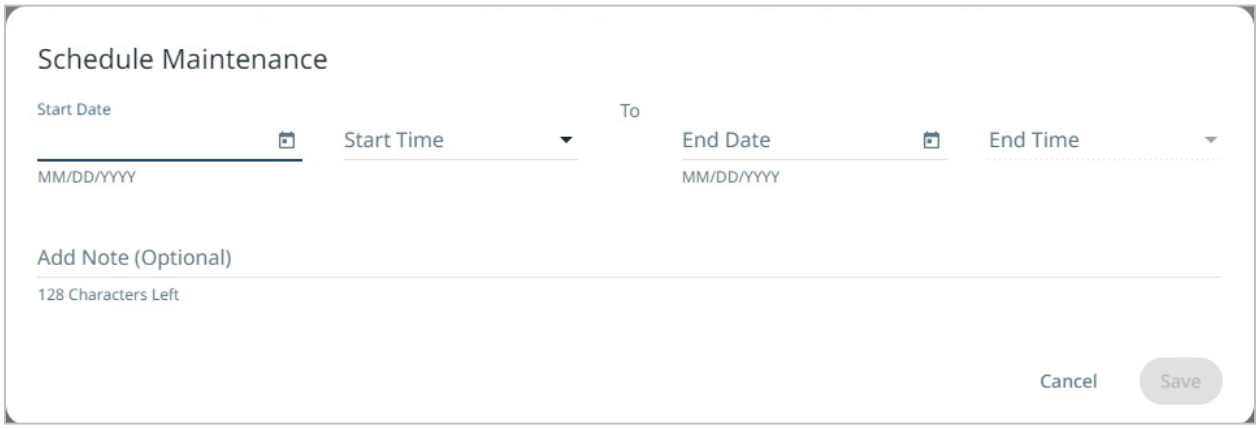


Registered Source Maintenance Mode

In large enterprises, thousands of Physical server sources are protected by Protection Groups. Sometimes, these sources need to be brought down for maintenance, such as OS downtime, hardware refresh, etc. The backup jobs scheduled for that period fail when a source is unavailable. Even if any recovery is attempted during that period, it fails. Customers facing these challenges often delete such sources from Cohesity to ensure backup will succeed, as pausing the protection group is not a suitable option because it contains other sources. The alternative is to live with error reports and alerts, which can have cascading effects based on customer workflows.

Cohesity introduced the Maintenance Mode feature for these sources to overcome the above challenges, reducing the customers' overhead. You can easily enable Maintenance Mode for registered sources. Enabling Maintenance Mode for your Cohesity cluster's registered sources will skip scheduled activities or manually initiated operations, such as protection runs, restores, and upgrades. Any activity, such as ongoing backups, will be canceled, but ongoing restores will continue without interruptions.

Schedule Maintenance



Bulk Deployment Methods

Nowadays, one of the major challenges customers have is the mass deployment of the Cohesity Physical Agent. Installing the Cohesity Physical Agent on each server requires a huge manual effort and administrative overhead. However, you can use the Cohesity dashboard to deploy the Cohesity Agent with ease.

Deploy using Cohesity Dashboard

The Cohesity dashboard allows software deployment with minimal or no user intervention. It supports bulk silent agent installation on servers, and registration on the Cohesity cluster. Use the following deployment methods for deploying agents from the Cohesity dashboard:

- vCenter deployment (Browse vCenter)
- Physical Server (Enter Hostname/IP)
- CSV Method (Upload CSV)

Based on your existing production infrastructure, Cohesity recommends one of the following three bulk deployment models.

Table 3: Bulk Deployment Model

Bulk Deployment Model	Recommendations
vCenter Deployment	You can select this method if you want to deploy the Cohesity agent on Virtual Machines, in a VMware environment.
Physical Server Deployment (Enter Hostname/IP)	You can select this method for deployment if you want to do the bulk deployment of Cohesity Agent on Windows and Linux servers using operating system tools.
Upload CSV Deployment	You can select this method if you have one of the following conditions, which need customization - • The target servers don't have a common user. • The password is not exact or identical for the target servers. • Different servers need different Cohesity Agent configurations. (For example, server A needs FSCBT, and server B doesn't need FSCBT.)

vCenter Deployment

This method lets you deploy the agent on a virtual machine of VMware vCenter or ESXi server and register it on the Cohesity cluster. If you must deploy the Agent on multiple virtual machines, use the vCenter deployment method. This allows you to:

- Perform efficient file-level recovery.
- Protect VMs as physical servers.
- Protect the application running on VMs.

Ensure the prerequisites are met before deploying the agents.

Prerequisites

- The required firewall ports are open to enable communication between the Cohesity cluster and the physical server (Virtual Machine).
- Minimum permissions required to deploy Cohesity Agent are in place.
- Windows UAC is configured as per the organization's security policy.
- Virtual machines have the latest version of VMware tools installed.
- Virtual machines should be up and running.
- vCenter and ESXi must be registered under Cohesity Sources.

Ports Required

For deploying Cohesity Agent on a virtual server, port 50051 must be open.

Required Permissions

- The user account used for auto-deploying agents in a Windows VMware environment must have administrator privileges on the VM on which the agent is deployed. The user account must be part of the administrator group.
- If Windows Firewall is active between the cluster and the target VM, we recommend adding an inbound rule to accept Cohesity Agent and TCP connections on local port 50051.
- The user account used for file recovery to a Windows VM must have administrator privileges on the target VM. The user account must be a part of the administrator group.

Deployment Considerations

- **Windows OS:**
  - By default, the Cohesity Agent will be deployed using the Local System account. If you need to use another user account, you can mention a specific user account during deployment.
  - By default, the deployment path is **C:\Program Files\Cohesity**, which can be customized as per the requirement.
  - **Volume CBT (Change Blocks Tracker):** This component is required to perform incremental backups in block-based mode. If you have registered the server as a physical server, Volume CBT will be installed by default. Cohesity recommends rebooting the server after installing this component.
  - **File System CBT (Change Blocks Tracker):** This component is required for the incremental backup of individual MS SQL databases hosted on a physical server registered as a file system. It is not installed by default. To install this file system CBT, click the Edit icon and enable the **File System CBT (Change Blocks Tracker)** option.
- **Linux OS:**
  - Cohesity Agent will be deployed using the `cohesityagent` user and group. During the agent deployment, the installer updates the system file `/etc/sudoers`.
  - You can create a new user by selecting "**Create the user if it does not exist**."
  - If you use an existing user for agent deployment, ensure it is added in the `/etc/sudoers`. The syntax is as follows:

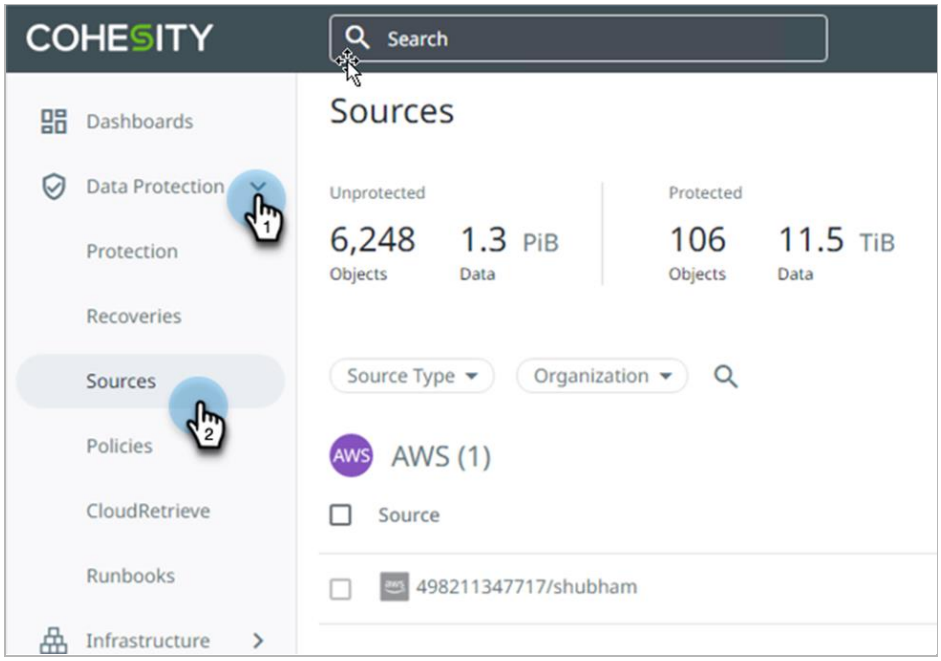
```
<user_name> ALL=(ALL) NOPASSWD:ALL
Defaults:<user_name> !requiretty
```

  - The default installation directory for Cohesity Agent is the "/opt/cohesity/agent" directory, which can be customized according to the requirements.
  - Required ports are open for the registration of servers on the Cohesity cluster.
  - `PasswordAuthentication` should be yes in the sshd_config file, or you can leverage ssh keys if you do not want to alter the sshd_config, you can leverage SSH keys.

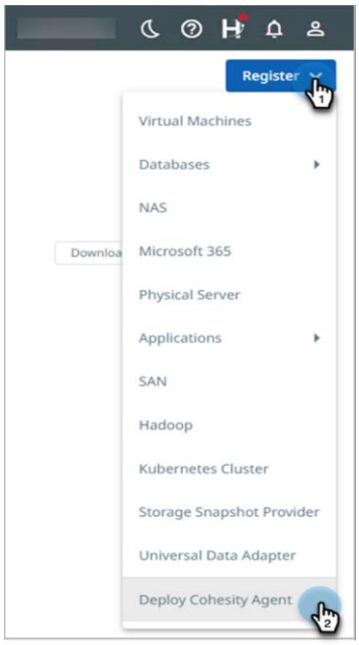
Deployment Steps

To deploy Cohesity Agent from the Cohesity dashboard on multiple virtual machines, follow the steps below:

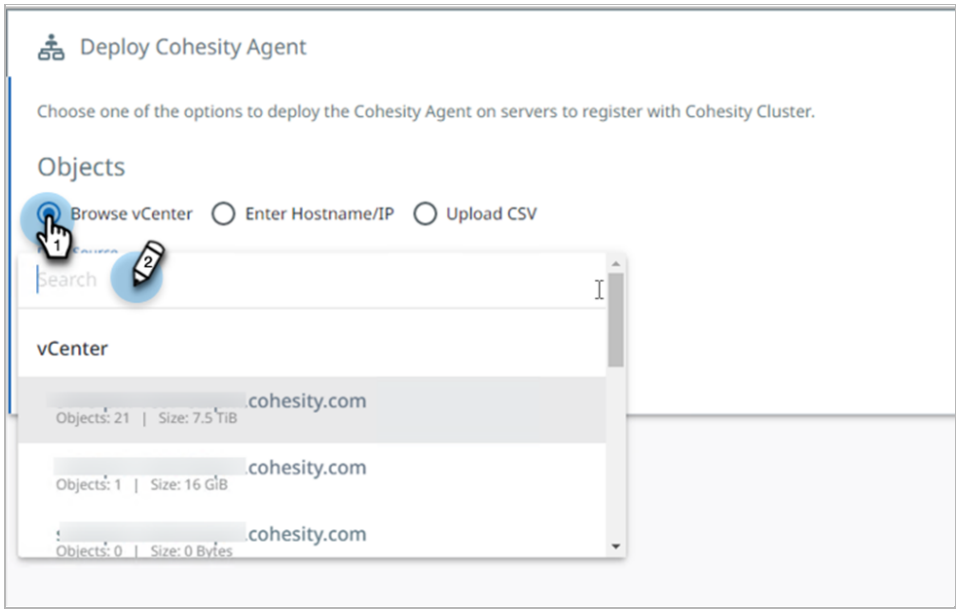
1. Login to the Cohesity cluster and navigate to **Data Protection > Sources**.



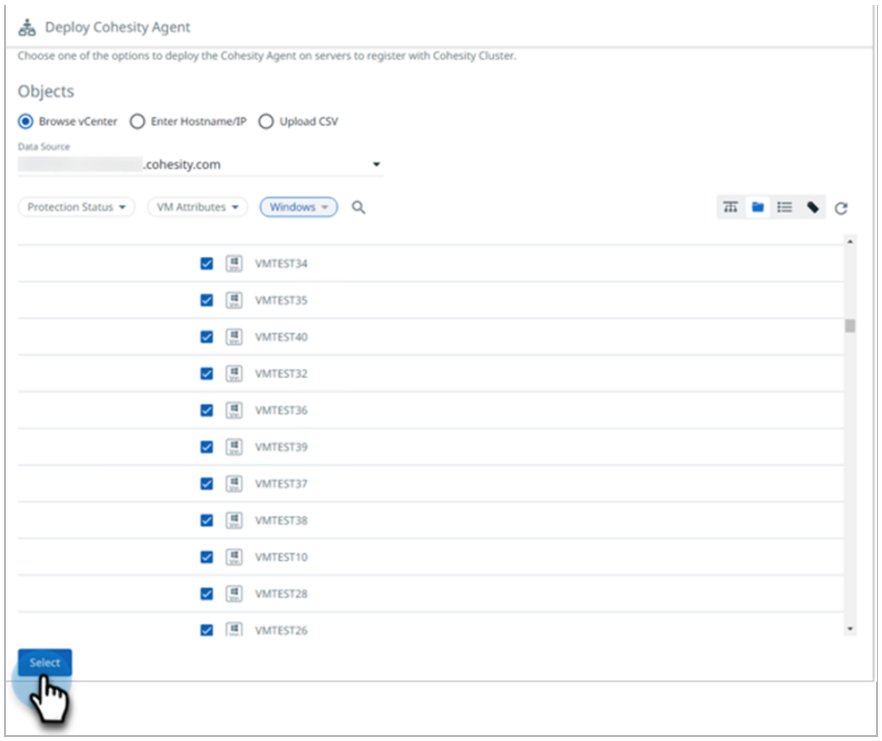
2. On the right pane, click **Register** and select **Deploy Cohesity Agent**.



3. Choose **Browse vCenter** and from the **Data Source** field, select the desired vCenter. Ensure that your vCenter is already registered with the Cohesity cluster.

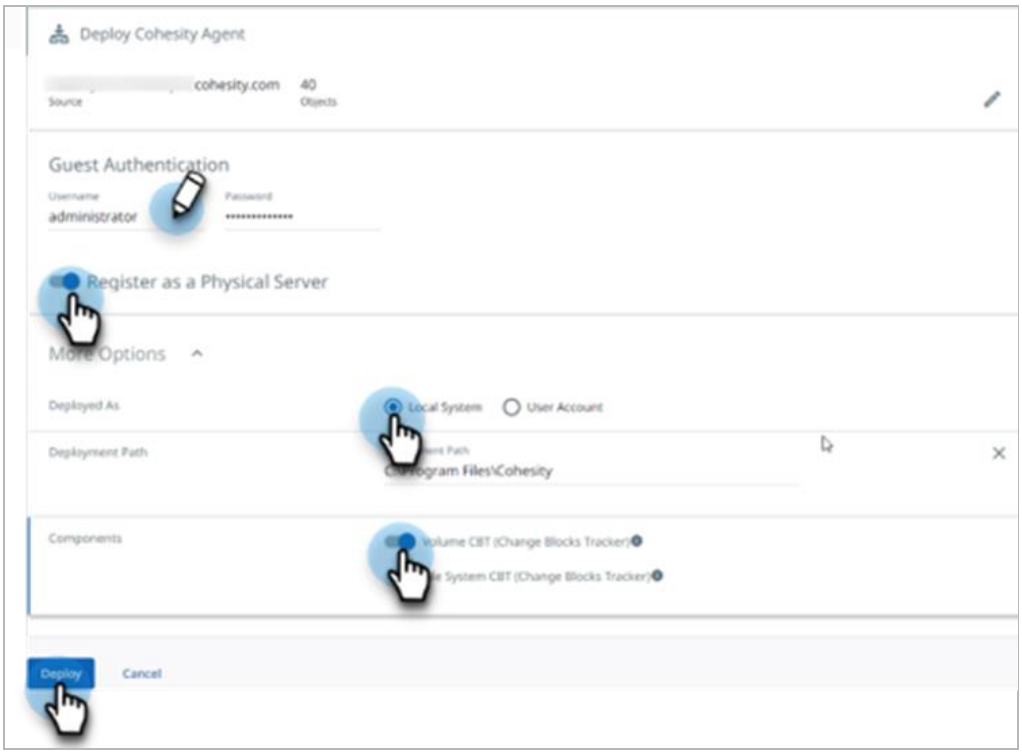


4. You can select the following objects to deploy Cohesity Agent for your operating system (either Windows or Linux). It will deploy Cohesity Agent on all the child entities.
   a. vCenter ( )
   b. vCenter Datacenter ( )
   c. Cluster Resource Pool ( )
   d. Virtual Machine ( )
   e. VMware Folder Resource ( )
   f. VMware Tags ( )
   g. Use the global search for specific text (Search)

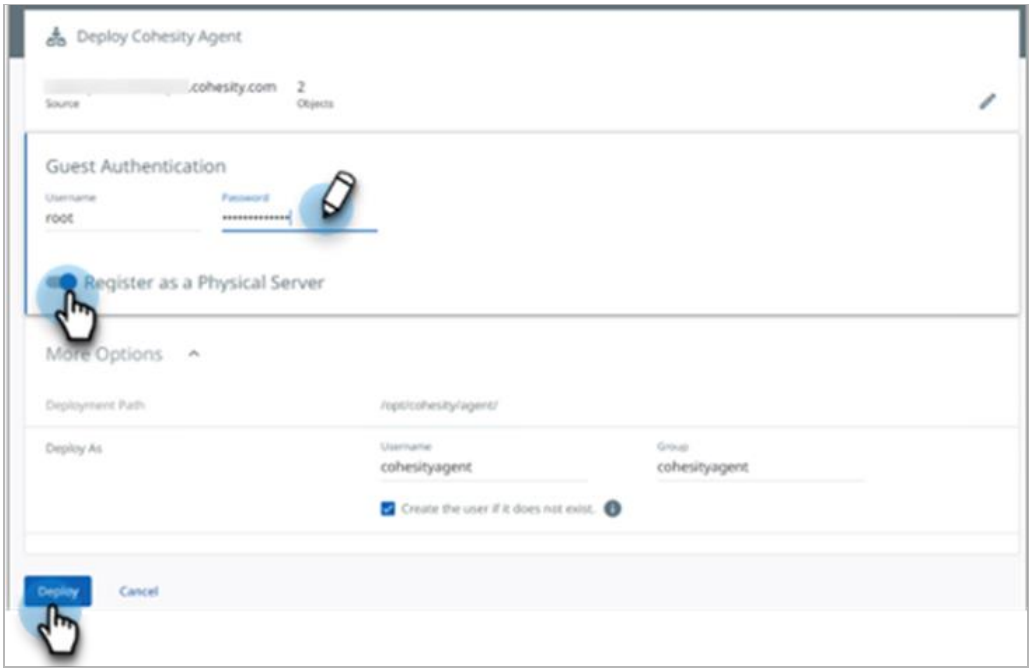


5. Click **Select**.
6. The next section will reflect the source and number of objects selected for the deployment of Cohesity Agent. Enter the following information (for more information, see the **Deployment Considerations** section).
   a. In Guest Authentication, enter **Username** and **Password**.
   b. Enable the option to **Register as a Physical Server**.
   c. Expand **More Options**.
      i. You can deploy Cohesity Agent by using the Local System as well as the Domain User Account (the Domain user account must have administrator privilege).
      ii. You can select or update the deployment path.
      iii. You can choose **Volume CBT** and **File System CBT** for deployment.
   d. Click **Deploy**.

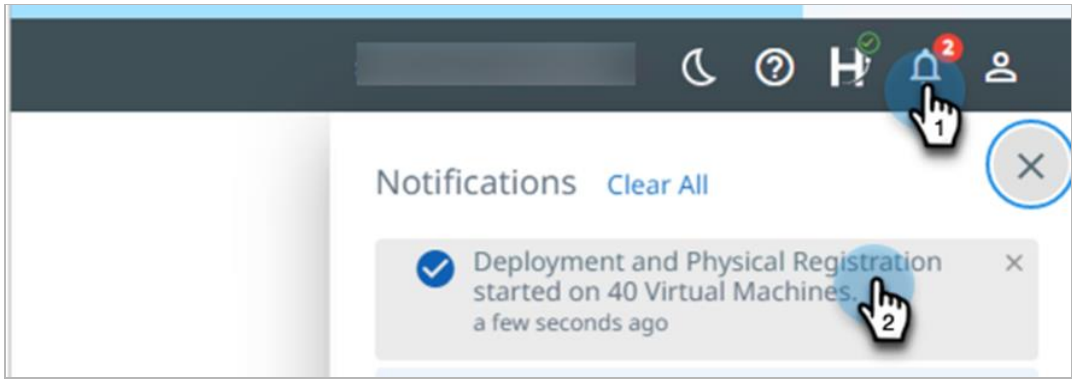
Windows Options:



Linux Options:

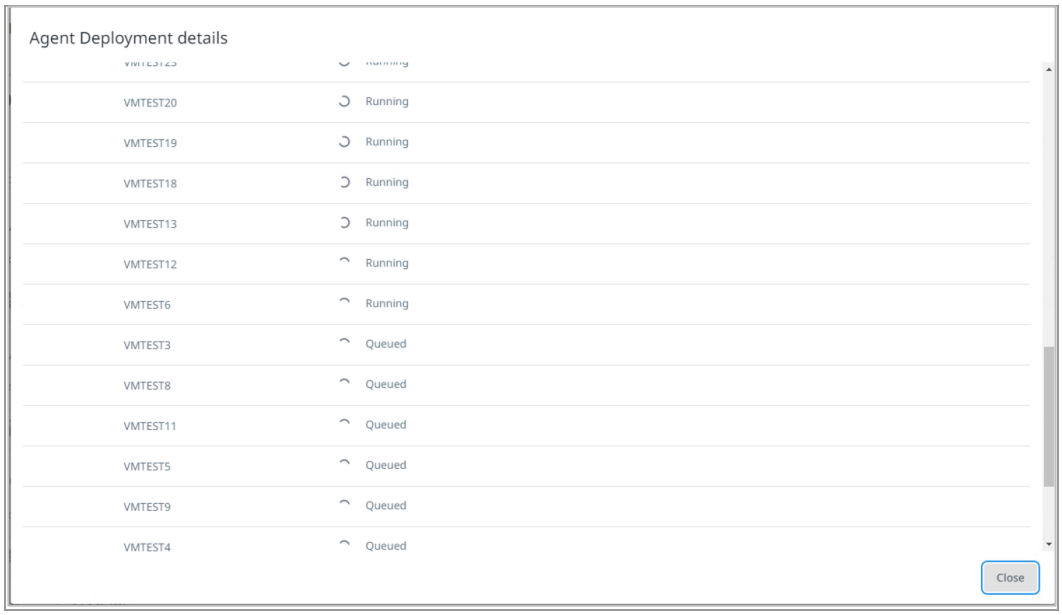


7. You can check the status of deployment by clicking the Notifications ( ) icon.

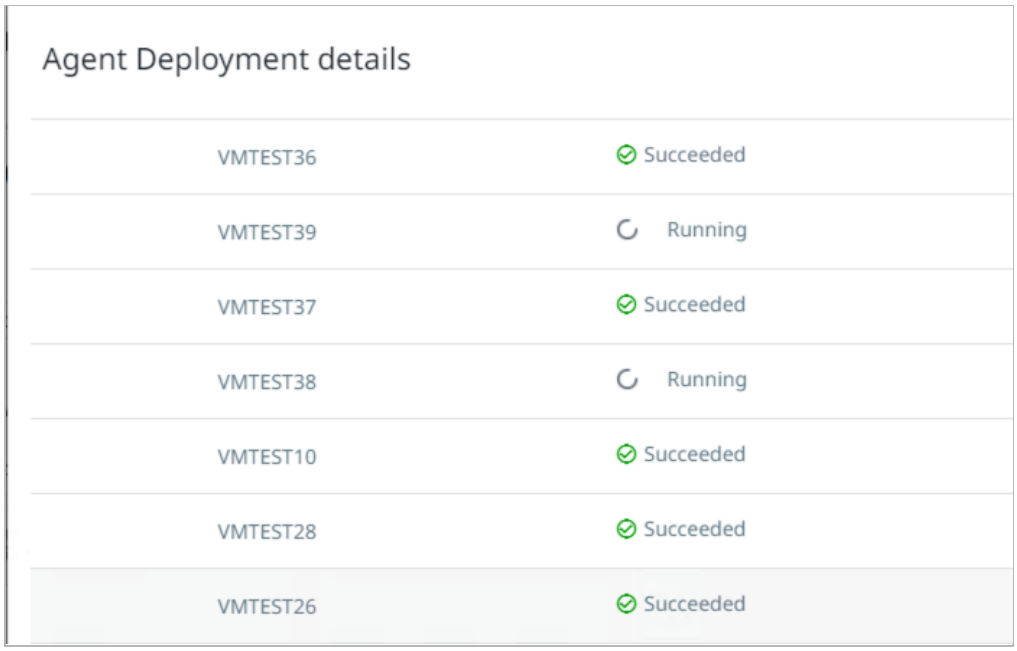


It will reflect the deployment status of the agent. The deployment process has the following four states.

- Running
- Queued
- Succeeded
- Failed



8. Once the agent is installed, the deployment status changes to **Succeeded**.



Physical Server Deployment (Enter Hostname/IP)

This section explains the deployment of Cohesity Agent on multiple servers by relying on the Hostname or IP Address of hosts having the same operating system and credentials. It will reduce the overhead involved in performing the installation administration of on every server. The benefits remain the same as before:

- Perform efficient file-level recovery.
- Protect VMs as physical servers.
- Protect the application running on VMs and physical servers.

Ensure the prerequisites are met before deploying the agents.

Prerequisites

- The Cohesity cluster and physical server should be able to communicate using the required ports through the firewall.
- Physical server user account should have the required minimum permission for the installation of the Cohesity agent.
- WinRM (SOAP) should be enabled on the Windows physical server.
- SSH service is running on the Linux physical server.

Ports Required

Ensure the following ports are open unidirectionally based on the operating system of the server to deploy the Cohesity Agent:

Table 4: Firewall Ports Requirement

Port	Usage	Windows	Linux
50051	To enable communication between the Cohesity cluster and Cohesity Agent running on the physical server.	Yes	Yes
22	To SSH to Cohesity Support Channel	No	Yes
5985/5986	WinRM HTTP (5985) or HTTPS (5986) connectivity	Yes	No

Minimum Permissions

Ensure you have the following permissions for deploying Cohesity Agent.

For Windows servers

- Use the **LOCAL SYSTEM** account or an account that meets the following requirements to install Cohesity Agent:
    - The account must be a member of the local Windows Administrators group. For example, `qa01\tse-backup` is an Active Directory user account in the data center. If the backup admin plans to use this account, `qa01\tse-backup` must be part of the local Windows Administrators group on the Windows server.
    - The account must have Log on as a service in the User Rights Assignment on the Windows server to deploy the Cohesity Agent. Refer to the link for more information.
- Verify and update WinRM restrictions:
    - Ensure WinRM/vSphere API is not blocked by the simple file-sharing mode:
      From the Windows server, go to **Start > Run > secpol.msc > Local Policies > Security Options** and set Network Access: Sharing and security model for local accounts to Classic – local users authenticate as themselves.
    - Ensure WinRM is not blocked by Windows Remote User Account Control:
      Configure the registry by going **to Start > Run > cmd** and pressing **Ctrl-Shift-Enter**.
      Enter the following command:

      ```
      reg add
      "HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\system"
      /v LocalAccountTokenFilterPolicy /t REG_DWORD /d 1 /f
      ```

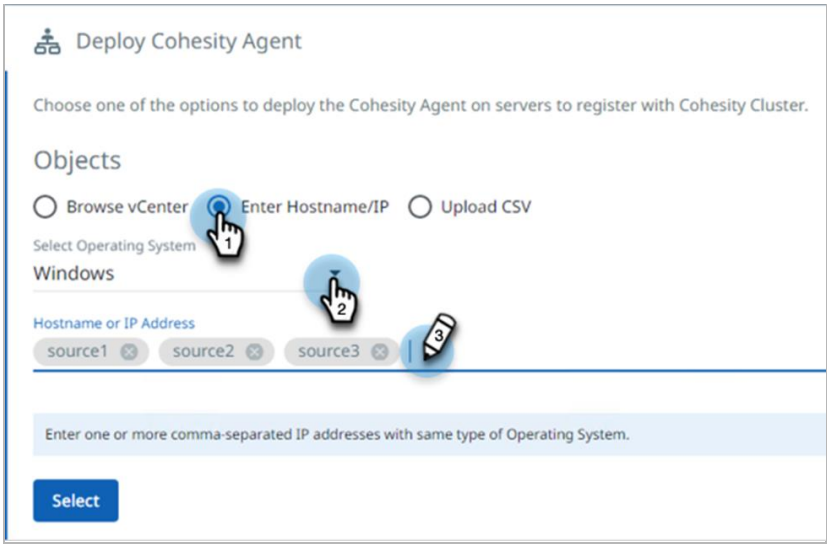
      For more information, see User Account Control.

Refer to the **Deployment Considerations** section for specifics related to the Linux operating system.

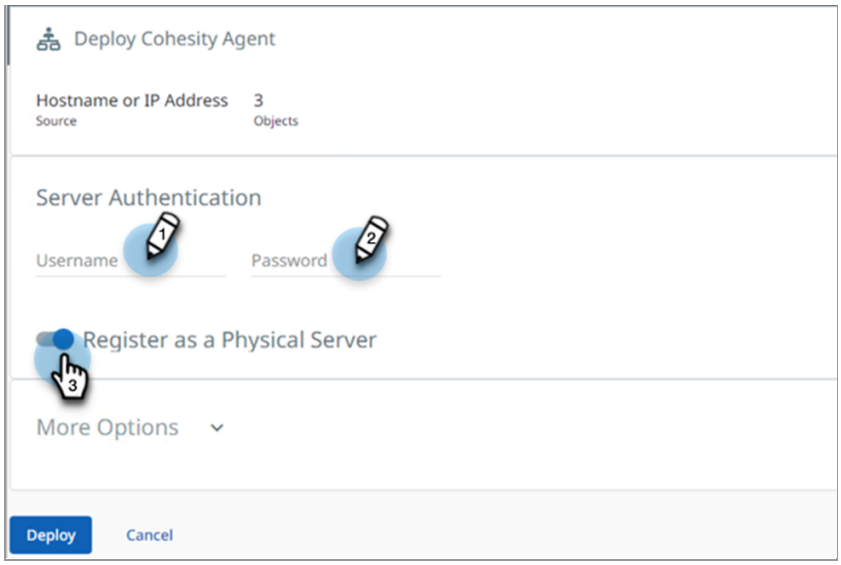
Deployment Steps

To deploy Cohesity Agent from the Cohesity dashboard by using the Hostname/IP, follow these steps:

1. Login to the Cohesity cluster and navigate to the **Data Protection > Sources**.
2. On the right side of the Web UI, click **Register** and select **Deploy Cohesity Agent**.
3. Select the Option **Enter Hostname/IP**:
   a. Select the Operating system, either Linux or Windows.
   b. Enter the Hostname/IP of the server. If you want to enter more than one of the same operating systems, then use the comma-separated hostname or FQDN of the source.



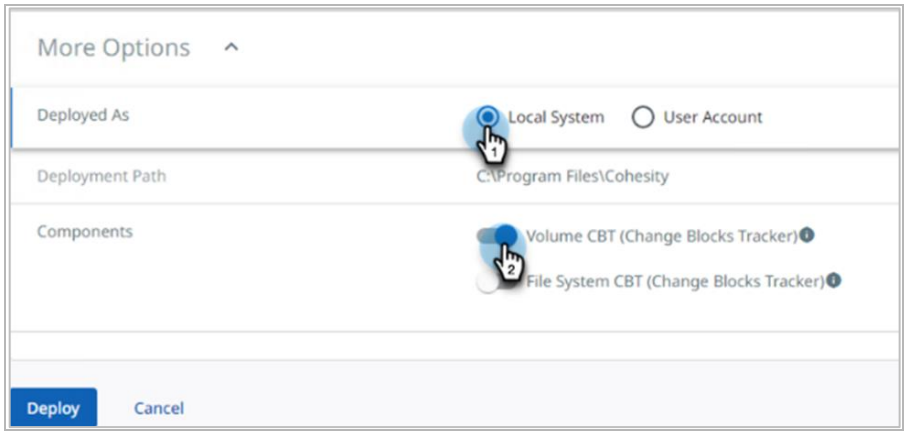
4. In **Server Authentication**, enter the username and password, which should be common for all the servers mentioned above.
5. If you want to register the servers as physical servers on the Cohesity cluster post-installation, toggle the **Register as a Physical Server** button.



6. Expand the **More Options** section:
   a. You can select or update the deployment path.
   b. You can deploy Cohesity Agent by utilizing the Local System or another user account. The user account should have local administrative rights and “Log on as a service”. This is applicable to Windows servers.
   c. You can choose **Volume CBT** and **File System CBT** for deployment. This is applicable to Windows servers.
7. Click **Deploy** to start the deployment.

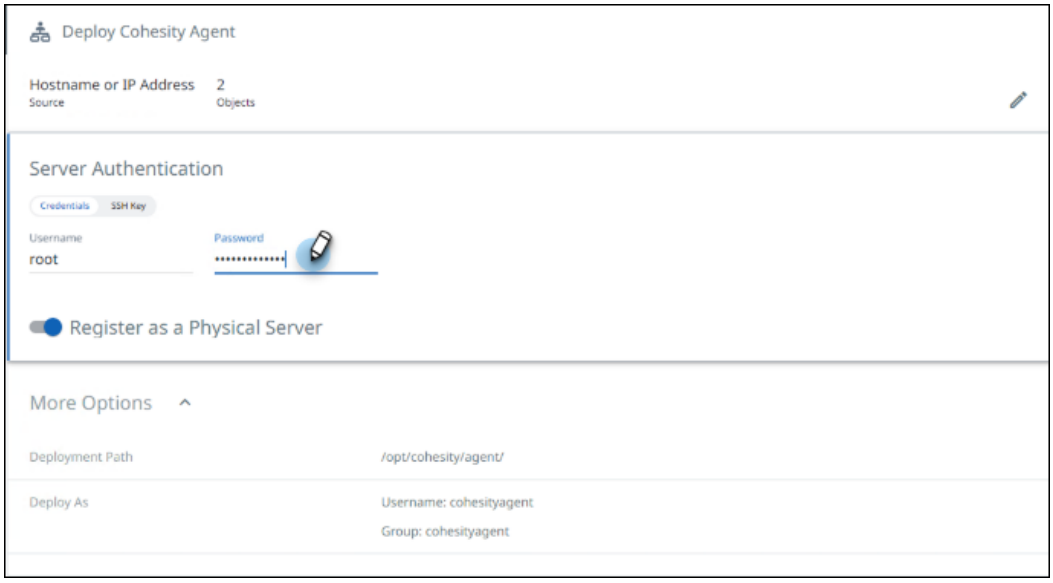
   Refer to the **Deployment Considerations** section for more information.

   Windows Options:



   Linux Options:

   To deploy the agent on the Linux operating system, you can either use a username/password, or pass the SSH Key.



8. To check the process and status, refer to step 6, step 7, and step 8 from **Deployment Steps** section under **vCenter Deployment** section.

Upload CSV Deployment

In this section, we will deploy the Cohesity Agent with customizations from the Cohesity dashboard. There can be situations where the target physical servers are not identical in terms of their configuration. For example, they may have different credentials, or a few servers need option A for agent installation, while others need option B. All these customizations are possible in this method. This section describes the mass deployment of Cohesity Agents by using the upload CSV option for non-identical systems.

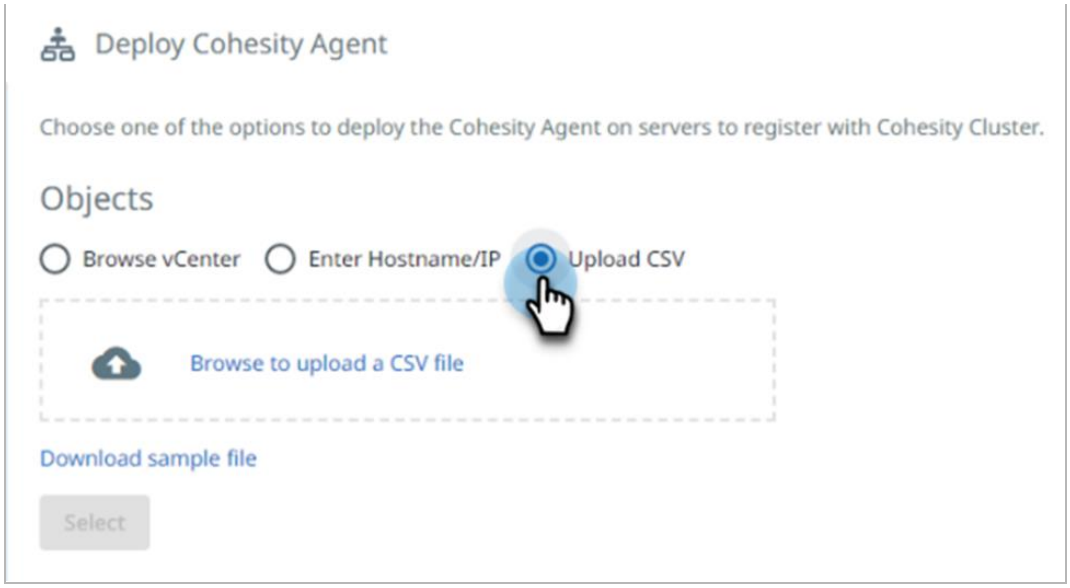
While deploying Cohesity Agents via the CSV method, you can target **both Windows and Linux operating systems** in one CSV file and deploy. In the two previous methods, we have the deployment limitation of not being able to process mixed operating systems.

NOTE: Prerequisites, port requirements, and minimum permissions are identical to **Physical Server Deployment (Enter Hostname/IP)** section.

Deployment Steps

To deploy Cohesity Agent from the Cohesity dashboard by using the CSV approach, follow the below steps.

1. Log into the Cohesity cluster and navigate to **Data Protection > Sources**.
2. On the right side of the page, click **Register** and select **Deploy Cohesity Agent**.
3. Select the option **Upload CSV** and then upload your prepared CSV file or download the sample CSV file.



4. The sample CSV file has the following 13 options. Enter the information related to the target servers as per the respective fields:

```
#IP/Hostname,OS Type,Server Username,Server Password,Install
Location,Volume CBT,File CBT,Service Account Username,Service Account
Password,SSH Key,linux user, linux group, create new user
```

The fields in the sample CSV are as follows:

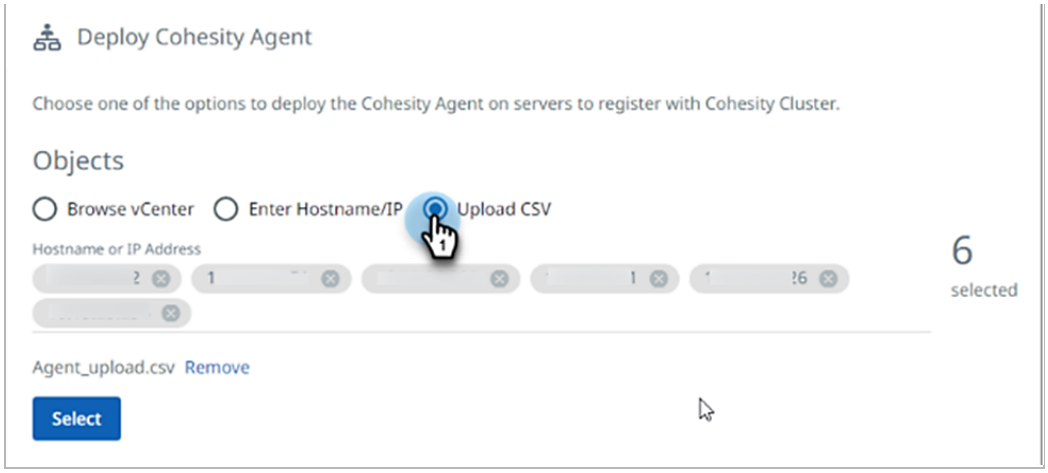
- **IP/Hostname**: The IP address or a valid hostname on which Cohesity Agent needs to be deployed.
- **OS Type**: The operating system of the server: Windows/Linux.
- **Server Username** and **Server Password**: The credentials of the server on which Cohesity Agent needs to be deployed.
- **Install Location**: (Optional) The path where Cohesity Agent must be deployed. If left blank, then the agent will be installed in the default location. Kindly mention \\ for Windows. For example C:\\Program Files\\folder\
- **Volume CBT**: (Optional) Applicable only for Windows OS types. If set to TRUE, the volume CBT driver will be installed. This field is ignored for non-Windows OS types.
- **File CBT**: (Optional) Applicable only for Windows OS types. If set to TRUE, the file CBT driver will be installed. This field is ignored for non-Windows OS types.
- **Service Account Username** and **Service Account Password**: (Optional) Credentials to register Cohesity Agent as a service account. This enables the agent to use credentials specific to the application users.
- **SSH Key**: (Optional) Use the SSH key to authenticate the server to deploy Cohesity Agent.
- **Linux User:** Applicable to Linux servers, this is the user required for installation if the default user account is not preferred.
- **Linux Group:** Applicable for Linux servers, this is the group required for installation if the default group is not preferred.
- **Create New User:** Enter the value as true or false if you want to create a new user which is mentioned in the Linux user and Linux group field.

Examples of filled CSVs are given below:

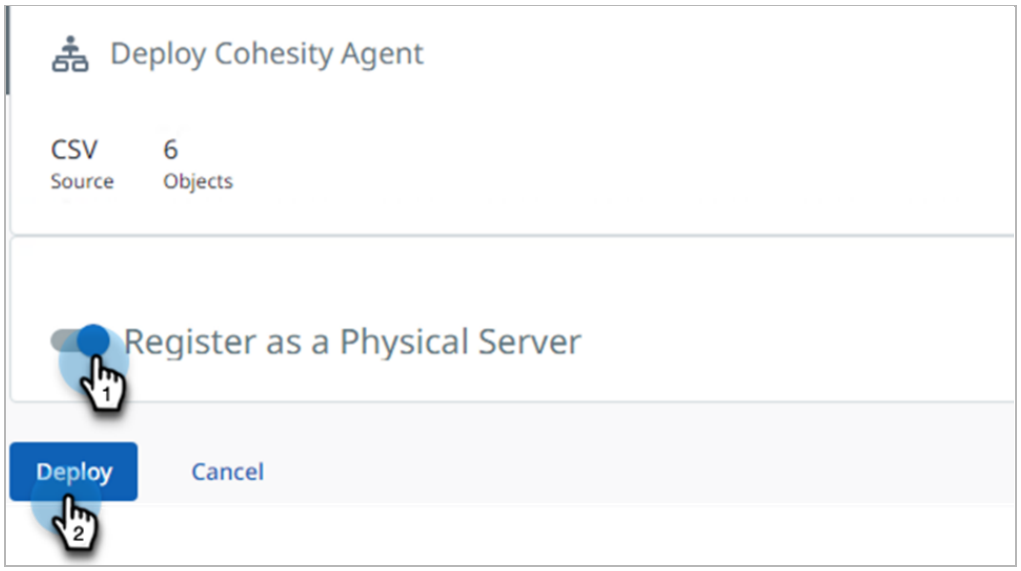
- 10.0.0.1, linux,admin,test_password,/var/agent_location/,,,,,,
- 10.0.0.2, windows,admin,test_password,C:\\Program Files\,,,service_admin,service_password,,

Refer to Appendix B: CSV Preparation before uploading the CSV for deployment.

5. Upload the CSV file to the Cohesity UI, which will then reflect the IP or FQDN in the **Hostname or IP Address** field:



6. Click **Select**.
7. Toggle on the option to **Register as a Physical Server** on Cohesity UI. Click **Deploy**.



8. To check the process and status, refer to step 6, step 7, and step 8 from **Deployment Steps** section under **vCenter Deployment** section.

Cohesity Agent Automation

The Cohesity Agent deployment methods described above allow you to install and register physical servers using the Cohesity UI. You can also use a standard process to perform the following tasks. To overcome the challenge of managing multiple servers, you can use scripting. This section describes automation scripts that are available here.

NOTE: This code is provided on a best-effort basis and is not in any way officially supported or sanctioned by Cohesity. The code is intentionally kept simple to retain its value as an example. The code in this repository is provided as-is, and the author accepts no liability for damages resulting from its use.

Cohesity does not offer any support for these scripts. You can use them and modify them accordingly.

Refer to the downloaded documentation before running the Powershell scripts.

Multiple Source Registration

To register a physical server on the Cohesity UI, you can refer to the documentation. You can register multiple physical servers with the Cohesity cluster once they have the Cohesity Agent installed. You can leverage this feature if you want to migrate the agents across the Cohesity clusters by following the below script.

1. Install Cohesity Agent before registration of the physical server.
2. Physical server should meet all the pre-requisite requirements as mentioned in **Cohesity Data Protection–Supported Physical Agents and Deployment Methodology** section.

Run the following command in PowerShell. It will download the script *registerPhysical.ps1* and the Cohesity REST API helper module *cohesity-api.ps1* from the relevant GitHub Repository into your current directory.

```
# Download Commands
$scriptName = 'registerPhysical'
$repoURL = 'https://raw.githubusercontent.com/bseltz-
cohesity/scripts/master/powershell'
(Invoke-WebRequest -UseBasicParsing -Uri
"$repoUrl/$scriptName/$scriptName.ps1").content | Out-File
"$scriptName.ps1"; (Get-Content "$scriptName.ps1") | Set-Content
"$scriptName.ps1"
(Invoke-WebRequest -UseBasicParsing -Uri "$repoUrl/cohesity-api/cohesity-
api.ps1").content | Out-File cohesity-api.ps1; (Get-Content cohesity-
api.ps1) | Set-Content cohesity-api.ps1
# End Download Commands
```

Place both the files together and run the script as follows:

```
./registerPhysical.ps1 -vip mycluster -username myusername -domain
mydomain.net -serverName w2016.mydomin.net
```

If you want to register a list of servers, use the following command:

```
./registerPhysical.ps1 -vip mycluster -username myusername -domain
mydomain.net -serverList ./servers.txt
```

You can use the following parameters for customization, as mentioned in the script:

Authentication Parameters

- -vip: (optional) name or IP of Cohesity cluster (defaults to helios.cohesity.com)
- -username: (optional) name of user to connect to Cohesity (defaults to helios)
- -domain: (optional) your AD domain (defaults to local)
- -useApiKey: (optional) use API key for authentication
- -password: (optional) will use cached password or will be prompted
- -mcm: (optional) connect through MCM
- -mfaCode: (optional) TOTP MFA code
- -emailMfaCode: (optional) send MFA code via email
- -clusterName: (optional) cluster to connect to when connecting through Helios or MCM

Other Parameters

- -serverName: name of server to register
- -serverList: text file containing list of servers to register (one per line)

Multiple Source Unregistration

To unregister a physical server using the Cohesity UI, you can refer to the documentation. To unregister multiple physical servers, you can follow the steps below which will only unregister the sources that are not part of any Protection Group. The unregistration process will report an error mentioning the Protection Group name for sources that are part of a Protection Group.

Run the following command in PowerShell, which will download the script *unregisterProtectionSource.ps1* and Cohesity REST API helper module *cohesity-api.ps1* from the relevant GitHub repository into your current directory.

```
# Download Commands
$scriptName = 'unregisterProtectionSource'
$repoURL = 'https://raw.githubusercontent.com/bseltz-
cohesity/scripts/master/powershell'
(Invoke-WebRequest -UseBasicParsing -Uri
"$repoUrl/$scriptName/$scriptName.ps1"). content | Out-File
"$scriptName.ps1"; (Get-Content "$scriptName.ps1") | Set-Content
"$scriptName.ps1"
```

```
(Invoke-WebRequest -UseBasicParsing -Uri "$repoUrl/cohesity-api/cohesity-
api.ps1"). content | Out-File cohesity-api.ps1; (Get-Content cohesity-
api.ps1) | Set-Content cohesity-api.ps1
# End Download Commands
```

Place both files in a folder together and run the main script:

```
./unregisterProtectionSource.ps1 -vip mycluster -username
myusername                                  -domain mydomain.net -sourceName
mysource1.mydomain.net, mysource1.mydomain.net
```

NOTE: Server names must exactly match the name reflected in the Cohesity UI.

You can use the following parameters for customization, as mentioned in the script:

Authentication Parameters

- -vip: (optional) name or IP of Cohesity cluster (defaults to helios.cohesity.com)
- -username: (optional) name of user to connect to Cohesity (defaults to helios)
- -domain: (optional) your AD domain (defaults to local)
- -useApiKey: (optional) use API key for authentication
- -password: (optional) will use cached password or will be prompted
- -noPrompt: (optional) do not prompt for password
- -tenant: (optional) organization to impersonate
- -mcm: (optional) connect through MCM
- -mfaCode: (optional) TOTP MFA code
- -emailMfaCode: (optional) send MFA code via email
- -clusterName: (optional) cluster to connect to when connecting through Helios or MCM

Other Parameters

- -sourceName: (optional) comma-separated list of source names to unregister
- -sourceList: (optional) text file containing source names to unregister

Cohesity Agent Upgrade on Multiple Sources

To upgrade Cohesity Agent on a single source, follow the steps mentioned in the documentation section. To perform the upgrade on multiple servers via the Cohesity recommended process, refer to **Cohesity** Agent Upgrade section, or leverage the PowerShell script to upgrade as below. Cohesity recommends using the procedure described in **Cohesity** Agent Upgrade section to upgrade Cohesity Agent.

This PowerShell script detects and upgrades Cohesity Agents that are older than the Cohesity cluster.

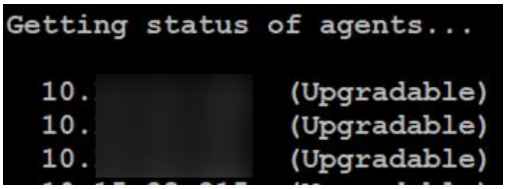
Run the following command in PowerShell, which will download the script *upgradeAgents.ps1* and Cohesity REST API helper module *cohesity-api.ps1* from the relevant GitHub Repository into your current directory.

```
# Download Commands
$scriptName = 'upgradeAgents'
$repoURL = 'https://raw.githubusercontent.com/bseltz-
cohesity/scripts/master/powershell'
(Invoke-WebRequest -UseBasicParsing -Uri
"$repoUrl/$scriptName/$scriptName.ps1").content | Out-File
"$scriptName.ps1"; (Get-Content "$scriptName.ps1") | Set-Content
"$scriptName.ps1"
(Invoke-WebRequest -UseBasicParsing -Uri "$repoUrl/cohesity-api/cohesity-
api.ps1").content | Out-File cohesity-api.ps1; (Get-Content cohesity-
api.ps1) | Set-Content cohesity-api.ps1
# End Download Commands
```

To get the list of physical hosts that are eligible for an agent upgrade, run the following script.

```
./upgradeAgents.ps1 -vip Cluster_vip -username cluster_username -domain
mydomain.net -all
```

The output will be as follows.



To perform the agent upgrade on all eligible physical hosts, run the following:

```
./upgradeAgents.ps1 -vip Cluster_vip -username cluster_username -domain
mydomain.net -all -upgrade
```

To perform the upgrade on specific physical hosts, run the following:

```
./upgradeAgents.ps1 -vip Cluster_vip -username cluster_username -domain
mydomain.net -serverName Host1.mydomain.net, host2.mydomain.net -upgrade
```

To perform the upgrade on a list of physical hosts described by a file called *Server_list*, run the following:

```
./upgradeAgents.ps1 -vip Cluster_vip -username cluster_username -domain
mydomain.net -serverList ./Server_list -upgrade
```

You can use the following parameters for customization, as mentioned in the script:

Parameters

- -vip: name or IP of Cohesity cluster
- -username: name of user to connect to Cohesity cluster
- -domain: your AD domain (defaults to local)
- -serverNames: one or more servers (comma separated) to report or upgrade
- -serverList: file containing list of servers
- -all: report or upgrade all servers
- -upgrade: perform upgrades (just report if omitted)

Cohesity Agent Upgrade

This section describes the supported methods to upgrade Cohesity Agents on physical servers. Cohesity recommends using the same version of a Cohesity Agent as is deployed on the Cohesity cluster. Keeping the agent version at the latest and compatible level will help users have an error-free backup. You can use the following methods for upgrading Cohesity Agents:

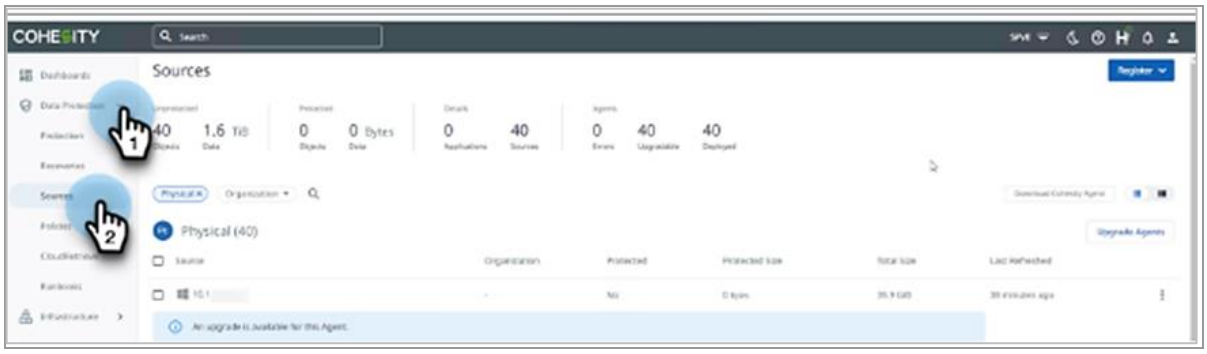
1. Cohesity Cluster dashboard
2. Individual Cohesity Agent Upgrade

Cohesity Cluster Dashboard

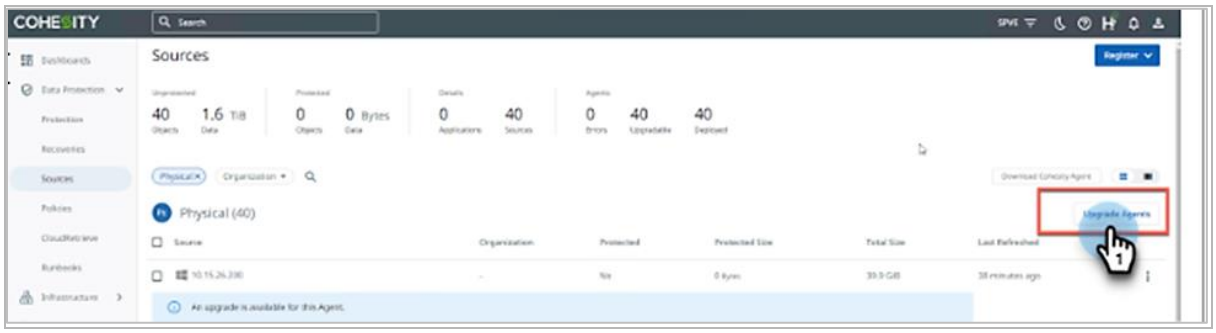
The Cohesity cluster dashboard provides information for all the agents available for upgrade. When applicable, you will see an information icon (ⓘ) on the sources page under the agent stating that **"**An upgrade is available for this Agent**".** To upgrade all the Cohesity agents available for an upgrade without minimum intervention, you can follow the below steps.

1. Navigate to **Data Protection** > **Sources**.

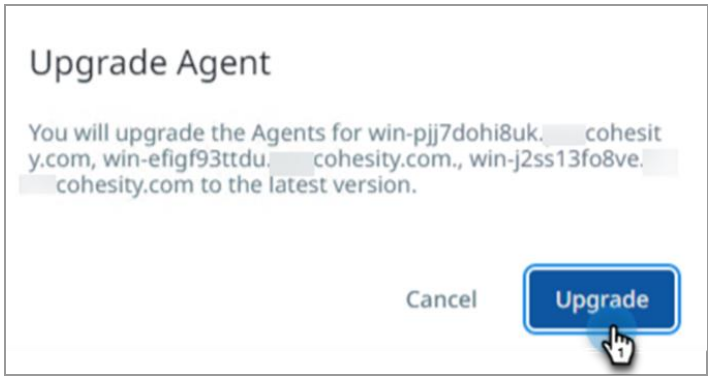
   Filter the sources as **Physical** and then you will see a button titled **Upgrade Agents** and information below each Cohesity Agent that is eligible for an upgrade. A summary of the number of eligible agents is also available on the dashboard.



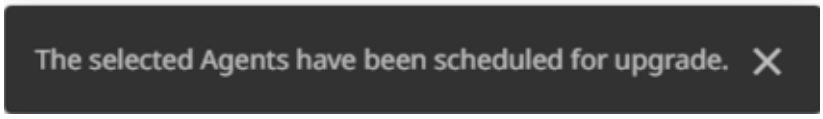
2. To see the list of all agents available for upgrade, click **Upgrade Agents**.



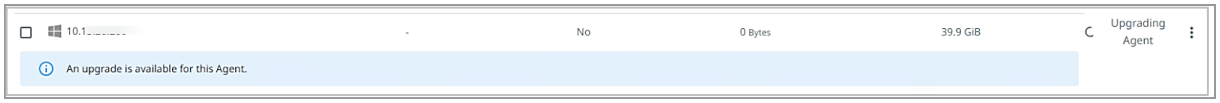
The resulting dialog lists all Cohesity Agents eligible for an upgrade –



3. Click **Upgrade**, and you will see the message on the bottom of the screen **"The selected agents have been scheduled for upgrade".**



The upgrade process will start for the applicable agents and will be reflected on the right side of the screen for each agent. Once the upgrade process is complete, each agent's information bar disappears.

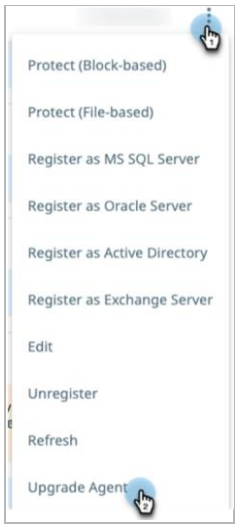


Upgrade Individual Cohesity Agent

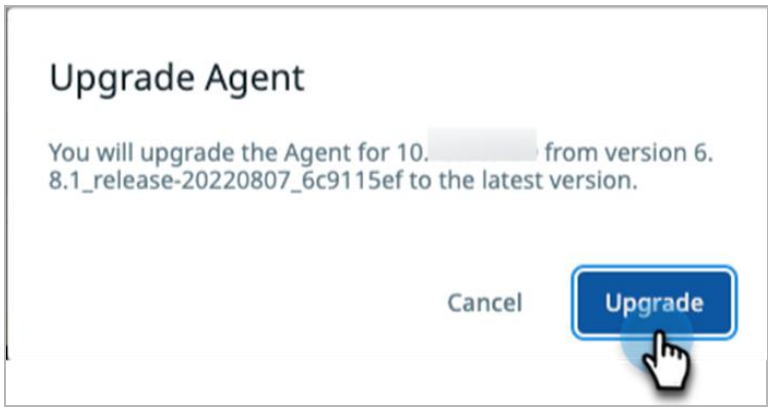
To upgrade a specific Cohesity Agent, you can manually select the agent from the sources tab and perform the upgrade process.

1. Navigate to **Data Protection** > **Sources**.

2. Click the Additional Options button (⋮) for the relevant agent and click **Upgrade Agent**.



3. An information window will appear that mentions agent information and the current version. If this looks correct, click **Upgrade**.



4. The upgrade process will start, and once the upgrade finishes, the additional information listed below the agent will disappear.

Cohesity Data Protection for Clusters

Protecting clusters is almost mandatory for enterprises as they are complex systems and run critical applications that can't afford downtime or disruption. Cohesity provides a cluster-aware solution to protect Windows failover cluster - File Server instances and Veritas Clusters so that no user intervention is required during the failover of cluster resources.

Windows Failover Cluster - File Server Role

A Windows failover cluster is a group of independent servers with shared resources that work together to provide high application availability. To have a unique identity, the cluster uses an IP address and cluster name. The clustered servers, called nodes, are connected such that in case one node goes down, then another node can take ownership and start serving the application.

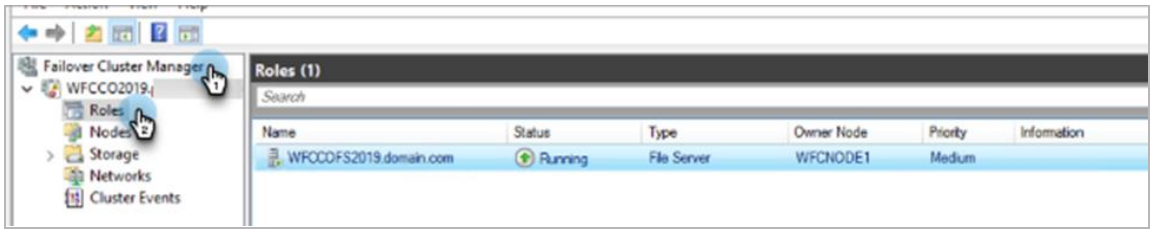
Windows failover clustering provides support for SMB shares and calls it the File Server role. The Windows File Server role allows customers to share files using attached storage and integrates seamlessly with Active Directory with NTFS permissions. Files are served to end users over SMB file shares that can be mapped as drives or accessed by UNC paths. Mapped drives can be easily deployed to users as they log in to their domain-joined Windows PCs through login scripts or group policy rules.

Prerequisites

1. Install Cohesity Agent on all the nodes of the Windows failover cluster.
2. Ensure that the Windows failover cluster nodes are identical in configuration.
3. Register with the VIP/FQDN of the file server role as a physical server on the Cohesity cluster.
4. Ensure that the Cohesity cluster is a member of Active Directory.

Protect Windows Failover Cluster - File Server Role for General Use

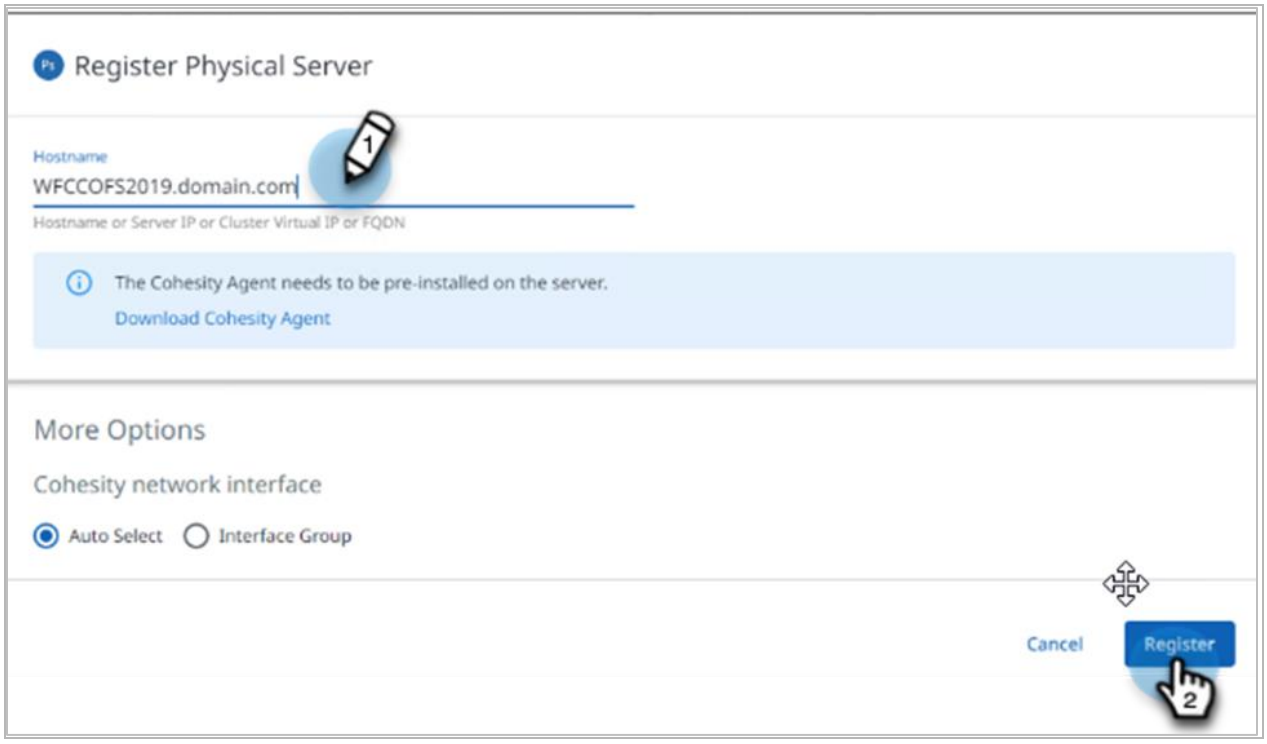
1. To get the Windows failover cluster's resource details, open the application *Failover Cluster Manager* and navigate to **Failover Cluster Manager** > **Roles**.



   Under the roles section, make a note of the file server role FQDN/VIP.

2. On Cohesity Cluster UI, navigate to **Data Protection** > **Sources** > **Register** > **Physical Server**.

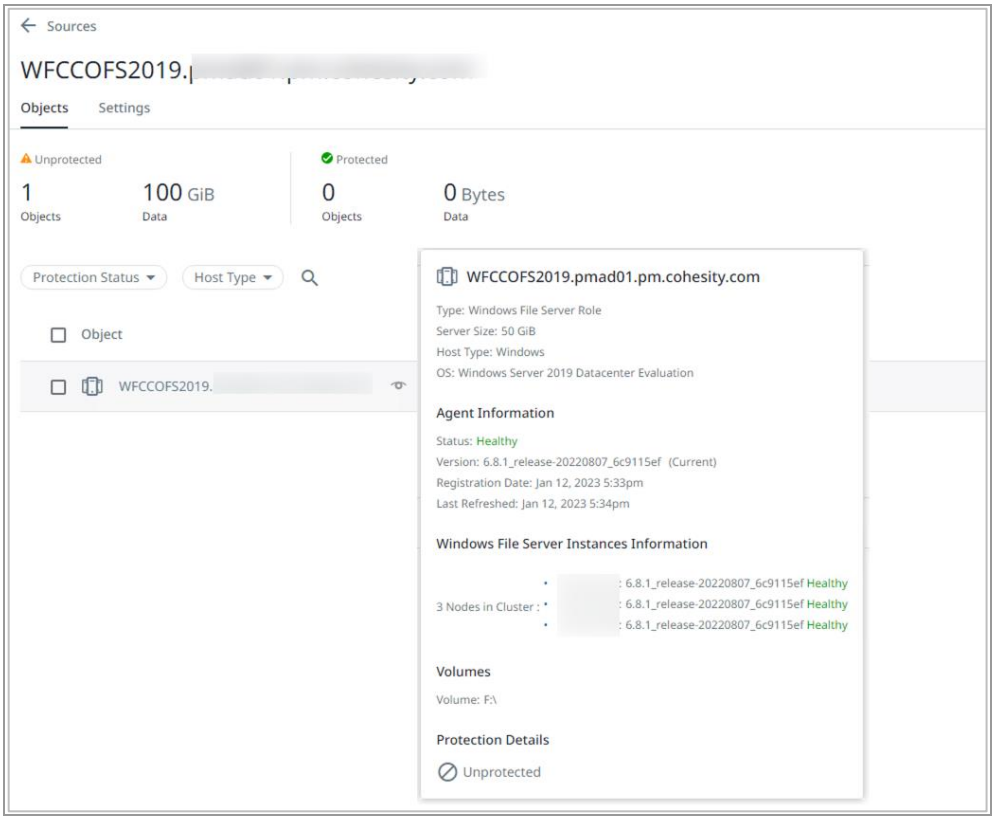
3. Enter the FQDN/VIP of the File Server role which you have noted in step 2 and click **Register**. It will register the cluster resource on the Cohesity cluster.



Post-registration, the cluster icon ( ) will be displayed for the registered Windows cluster source.

4. From the sources page, click on the FQDN/VIP of the File Server role of Windows cluster resource and click the eye icon. A pop-up window will be displayed, which will display the following information about the file server role:

   a. Type: Windows File server Role
   b. Server Size: shared disk part of cluster resource
   c. Host Type
   d. OS (Operating System)
   e. Agent information
   f. Volumes: Shared volume configured with File Server role



Cohesity provides two methods, blocked-based and file-based, to protect the Windows failover cluster - File Server role for general use. Cohesity recommends using the file-based data protection level in case the share does not have millions of files and the change rate for files is not high. File-based backup provides the following configuration options:

1. Follow Symlink NAS target
2. Protect all local volumes
3. Use directive files for backup

Block-based protection can be leveraged when the cluster-shared disk contains millions of files, or when there is a high change rate for the files involved.

File-based Protection

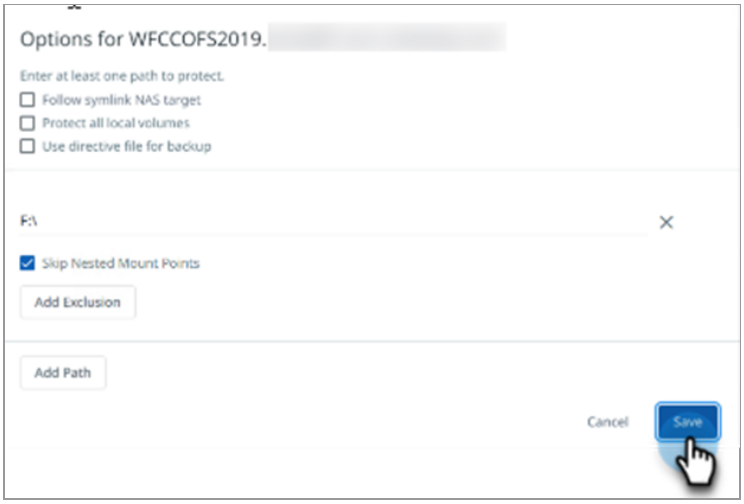
To protect the Windows failover cluster - File Server role by using the Cohesity file-based protection method, follow the steps below:

1. Navigate to **Data Protection** > **Sources** and click on the registered instance of the WFC-File Server.
2. Click the additional options button (⋮) on the right side of the Cohesity UI. Click **Protect File-based**.

3. On the Protection Group page, click the edit option next to **Add Objects**. Edit the object by clicking the  icon. This will allow you to select the following options:
   a. Follow symlink NAS target.
   b. Protect all local volumes.
   c. **Directive File for Backup**
   d. **Following is the** example for wildcards in directive files.

Directive File Contents	Directory Structure	What is included
/A/B/file? /A/B/*.log	/A/B/file1 /A/B/file2 /A/B/file10 /A/B/dir1 /A/B/info.log /A/B/error.log	/A/B/file1 (matches /A/B/file?) /A/B/file2 (matches /A/B/file?) /A/B/info.log (matches /A/B/*.log) /A/B/error.log (matches /A/B/*.log)

   e. Inclusion and Exclusion: Provide the file path in Exclusions to exclude any files/folders/drives from the backup.



4. Click **Save**. Select a predefined policy or create a new policy and click **Protect**. For more information on the creation of a protection group, refer to Protect a Physical Server (File-based).

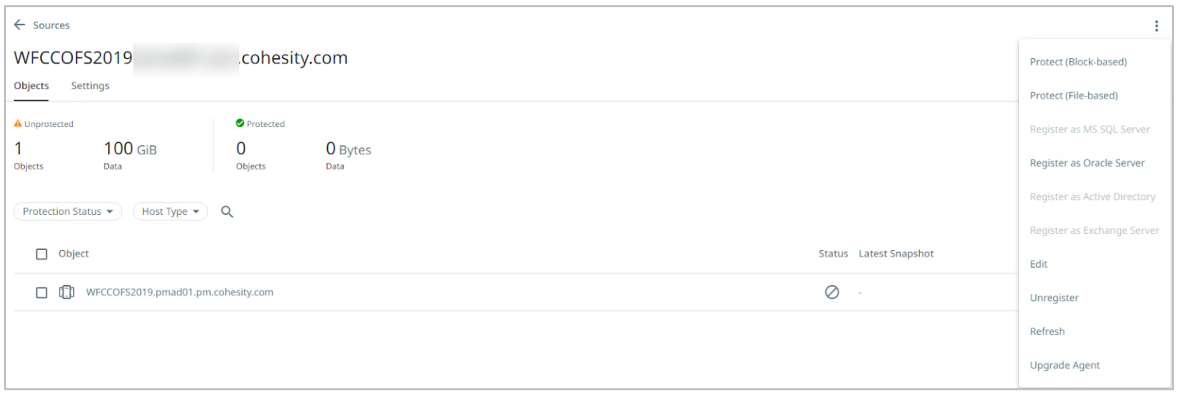
NOTE: You must create shares for the volumes to be discovered as WSFC File Server role volumes on the Cohesity cluster. Another point to note about incremental backups:

- For protection with **Crash Consistency enabled**, if there is no failover, all backups run in the incremental mode for the Windows cluster's shared disks. If there is a failover/failback on a shared disk over to the other node, the next backup will be full.
- For protection with **Crash Consistency disabled**, all backups are incremental even if there is a failover/failback of shared disks.

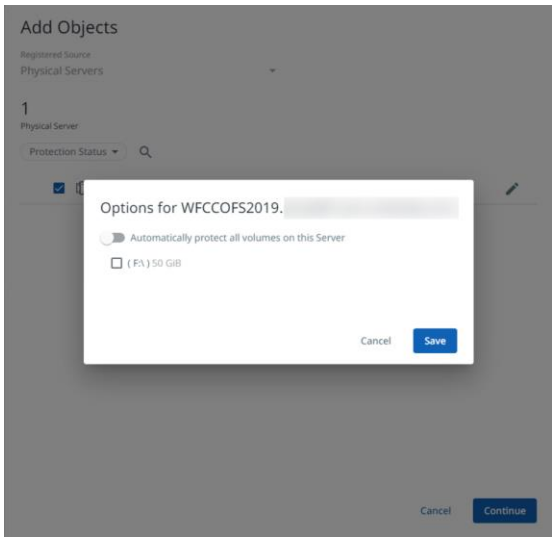
Block-based Protection

To create a Protection Group using the block-based backup approach, follow these steps:

1. Navigate to **Data Protection** > **Sources** and click on the registered instance of the Windows failover cluster - File Server role.
2. Click the additional options button (⋮) on the right side of the Cohesity UI. Click **Protect-block based**.



3. On the Protection Group page, click the edit option next to **Add Object** and edit the object. You can automatically protect all volumes on this server or select a specific volume.



Select the pre-defined policy or create a new policy and click **Protect**. For more information on the creation of a protection group, refer to Protect a Physical Server (Block-Based).

NOTE:

- If there is no failover, all the backups are incremental (leveraging CBT) for the shared disks. The next backup will be full if there is a failover/failback on a shared disk over to the other node.
- It would be best if you created shares for the volumes to be discovered as WSFC File Server role volumes on the Cohesity cluster.

Considerations

Review the following considerations before protecting a WSFC File Server role using Cohesity:

- Cohesity supports WSFC running on Windows Server 2012, Windows Server 2012 R2, 2016, 2019, and 2022 versions.
- WSFC File Server role protection is supported only on on-prem deployments. Data Management as a Service (Backup as a Service) and multi-tenancy are not supported.

- This feature supports backup and recovery of data associated with the File Server role in general use.
- For a crash-consistent backup, the backup will fail if there is a failover during backup.
- A standalone physical server Protection Group can also protect all volumes associated with the File Server role. However, Cohesity does not recommend protecting volumes associated with the File Server roles through a standalone physical source.
- Bare Metal Restore (BMR) is not supported for WSFC roles. However, BMR can be done on the Windows Servers which are part of the Failover Cluster.
- Upgrading the Cohesity Agent on a File Server role registered as a source will also upgrade the agents on all the nodes within the cluster.
- Backups will run through the owner node of the File Server role cluster as the cluster disks are offline and in a reserved state on other nodes.

Veritas Cluster Services

Veritas Cluster Server (VCS), bundled with the Storage Foundation and High Availability (SFHA) product from Veritas Infoscale Availability, is a high-availability cluster software for Unix, Linux, and Microsoft Windows computer systems. It provides clustering capabilities to servers running other applications over a file system.

It connects multiple independent servers into a management framework for increased availability. Each server or node runs its operating system and cooperates at the software level from a cluster. VCS links hardware and software layers to provide application failover and control. When a node or a monitored application fails, other nodes can take predefined actions to take over and bring up services on another cluster node.

Cohesity provides a solution to protect Veritas Cluster instances. File-based Protection Groups protect the data of the shared volumes of the cluster and run an uninterrupted backup post-failover/failback without manual intervention.

Prerequisites

- Refer to the documentation for the supported versions
- One node in the VCS cluster must be in vxfs active-passive configuration.
- Each node in the VCS cluster must have its own service group and each service requires access to all shared volumes.
- The same version of the Cohesity Linux Agent must be installed on all the nodes in the VCS cluster.
- The VCS cluster must be registered as a physical server using the VIP/FQDN.
- The following commands must be available on all the nodes in the VCS cluster:
    - `hares`
    - `hagrp`
    - `hasys`
    - `haclus`
    - `Lltconfig`

Veritas Cluster Server Registration

To register a Veritas Cluster instance with a Cohesity Cluster, use the VIP/FQDN of the VCS cluster name.

1. Run the following command on any of the nodes of the target VCS cluster to get the cluster name, which will be required for registration with the Cohesity cluster.

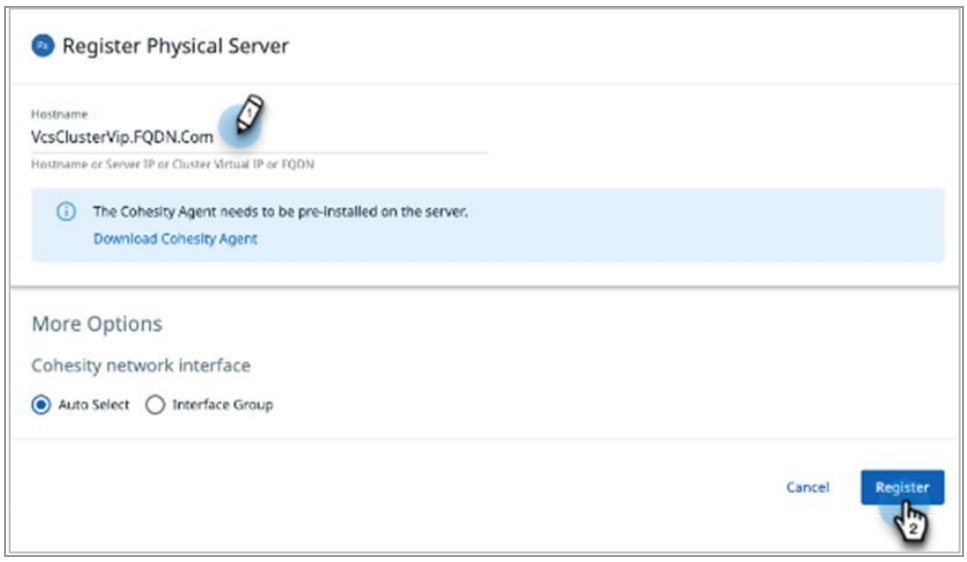
```
hares -display cvm_clus -attribute CVMClustName
```

```
[root@        /]# hares -display cvm_clus -attribute CVMClustName
#Resource     Attribute              System     Value
cvm_clus      CVMClustName           global     vcsdemo
```

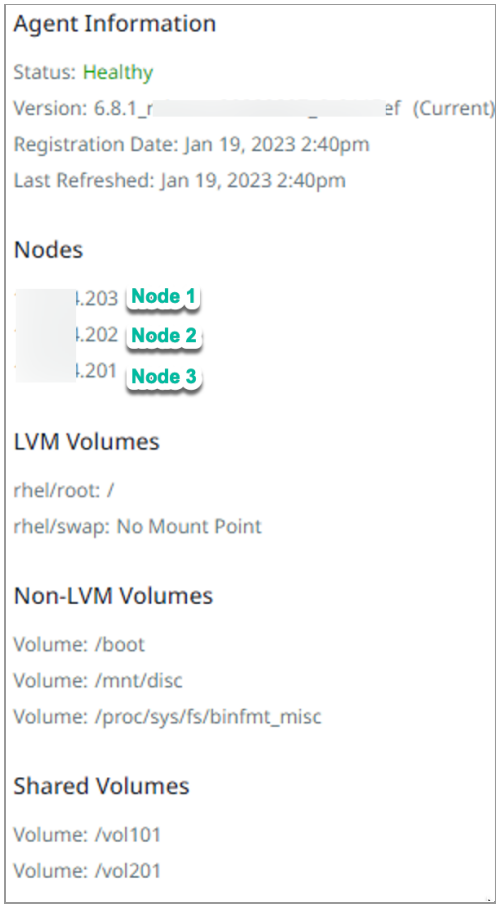
   From the above output, `vcsdemo` is the VCS cluster name. Ensure that the VCS cluster name is resolvable by DNS.

2. To register the VCS cluster VIP/FQDN, navigate to **Data Protection** > **Sources** > **Register** > **Physical Server** on the Cohesity UI.

3. Enter the FQDN/VIP of the Veritas Cluster Server, which you noted in step 1, and click **Register**.



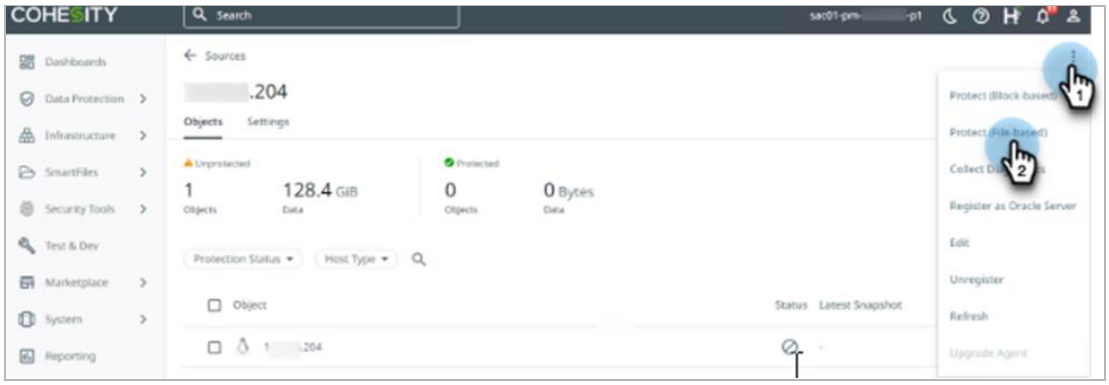
4. After the registration of the VCS cluster, the details of the VCS cluster configuration will be reflected in the UI, along with the cluster storage resource and its nodes.



File-based Protection (Veritas Cluster Server)

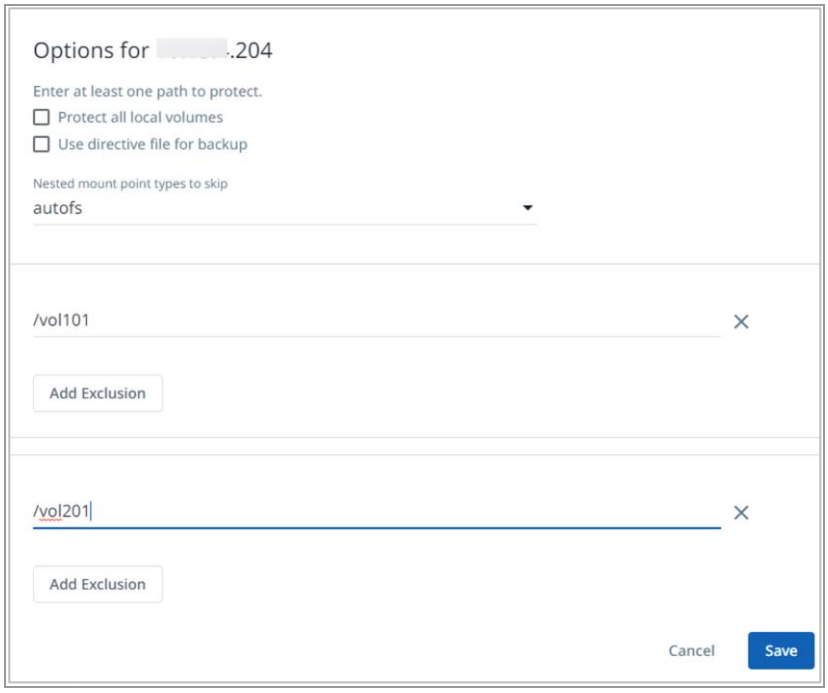
Cohesity recommends using file-based data protection for Veritas Cluster Servers. To create a file-based protection job for VCS, follow the steps below:

1. In the Cohesity UI, locate the source instance of the Veritas Cluster server, click the more icon (⋮), and then click **Protect (File-Based)**.



2. On the new protection page, edit the object ( ) and enter the information of the shared volume. You can also get this information from the Cohesity Cluster UI VCS source. Click the eye icon next to the source and it will reflect the shared volume information.

   You must enter the information of the cluster shared storage by registering the FQDN/VIP of the Veritas Cluster Server. If you want to protect the local volumes of the cluster node, then register the cluster nodes individually.



3. Enter the Protection Group name, select a predefined policy, or create a new one, and click **Protect**. For more information on the creation of a Protection Group, refer to Protect a Physical Server (File-Based).

Considerations

- You have to configure file-based backup and recovery of a VCS cluster running on a physical server.
- Cohesity supports Vxfs active-passive VCS cluster configuration.
- If you upgrade Cohesity Linux Agents of the VCS cluster from the Cohesity cluster using the VIP/FQDN, then only the Agent on the node to which the VIP is pointing will be upgraded. Other nodes can be upgraded by one of the following methods:
  - Add individual nodes to the Cohesity cluster as physical servers and upgrade the Cohesity Linux Agents as usual.
  - Initiate a failover of the VIP to other nodes and then upgrade the Cohesity Linux Agent on each node turn by turn.
- To protect VCS shared disks, use VIPs to register the VCS cluster, and to protect local disks, register the VCS with private IPs or VCS cluster node IPs.

- If a failover happens during backup, then the backup of some files might be skipped. These skipped files will be backed up in the next Protection Group run.
- If the failover takes more than 10 minutes, the Protection Group run might fail.
- If a failover happens during restores, a delay of up to 10 minutes can be tolerated. If the failover takes more than 10 minutes, then the restore will fail.

Protection Group Features & Configuration

To protect a physical server with a Cohesity Agent, you need to create a Protection Group, and add your physical source to it. You can have one or more physical sources within a Protection Group. Protection Groups provide multiple features that simplify the task of protecting physical sources. Protection Groups offer the following features.

Directive File for Backup

The directive file defines the location of the files and folders you can back up in a Protection Group for specific physical servers. You need to specify the file path of the directive file on the source. You should create this file, place it on the physical server, and then grant read-write access to Cohesity to parse it during the protection job run.

Each physical server should have its own directive file. While creating the protection job, select the '*Use directive file for backup*' check box and specify the file path of the directive file in the text box. The system will treat this as a directive file and parse every entry within the file to back up the data from all the defined locations as specified in the directive file.

For any reason, if data is not present at any of the specified paths, the protection job skips the entry and proceeds with the remaining entries within the file. After the protection job is complete, the system generates an **<input metafile>-out.txt** log file that contains the entries of all the successfully backed-up files. You can also find the location of this file in the protection job logs after the backup run.

Wildcards and exclusions (local and global) were not supported for directive file-based backups (till version 7.0). From Version 7.1, you can leverage wildcards. This functionality is available only for file-based backups on Windows, AIX, and Linux servers. Directive file backups are not supported on RHEL 5.

Following are the examples for Windows and Linux Directive files:

Entries in Windows directive file

- C:\Users
- D:\Departments\Finance
- D:\Departments\HR
- G:\Misc\department_list.txt
- G:\Misc_2\department_other_list.txt

Entries in Linux or UNIX Directive file

- /home/cohesity/meta_file.txt
- /home/cohesity/backup_folder

NOTE:

1. Cohesity recommends limiting the maximum number of paths in directive files to 10,000.
2. If you need to take a backup involving directive files, which contain more than 10,000 paths, then you should create a new directive file and create a new protection group with the physical server.

Following is the example for wildcards in directive files.

Directive File Contents	Directory Structure	What is included
/A/B/file? /A/B/*.log	/A/B/file1 /A/B/file2 /A/B/file10 /A/B/dir1 /A/B/info.log /A/B/error.log	/A/B/file1 (matches /A/B/file?) /A/B/file2 (matches /A/B/file?) /A/B/info.log (matches /A/B/*.log) /A/B/error.log (matches /A/B/*.log)

Inclusion and Exclusion

Cohesity provides the functionality of inclusion and exclusion of files and folders in file-based physical server backup, allowing users to choose the data to include and exclude. By default, all files and directories under the parent directory specified in the Include path of a protection group policy are protected by the Cohesity cluster, and the virtual file systems are excluded.

Furthermore, wildcards can be used in the Exclude paths in directories to advance and maximize the exclusion behavior. Cohesity, by default, supports only one wild character per Exclude path. However, for exceptional cases, support for more than one wildcard can be enabled to support regex wildcards. You can leverage the "?" and "*" to match the file for inclusion and exclusion. Entry should be a combination of a fully qualified directory path and supported wildcards (?,*) in the filename and has to be applied on the leaf level.

You can exclude the mount path by mentioning the absolute path. Wildcards are not supported for mount paths. For example, to exclude /home, specify the absolute path as /home; a wildcard pattern such as /hom* is not supported. Also, between exclusion and inclusion - Inclusion takes precedence over exclusion for local drive backup.

Cohesity supports the following wildcards:

Character	Description & Usage
Question mark (?)	• Represents any number of characters • Used for files only • Can be used only in the filename and not in the file extension • Cannot be used at the end of a string

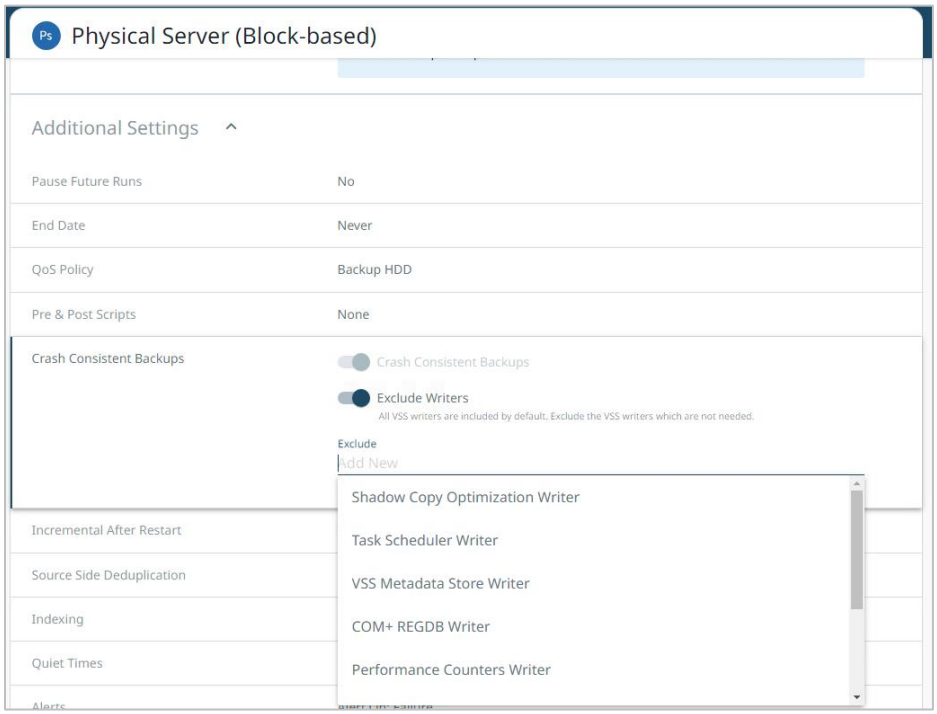
Character	Description & Usage
	• Scope is limited to the specified directory • Matches a single character of any type
Asterisk (*)	• Represents any number of characters • Used for both directories and files names • Can be used anywhere in the string • Scope is across all the directories and files in the Include path

For more details of Exclusion and inclusion refer to the Add Exclusion.

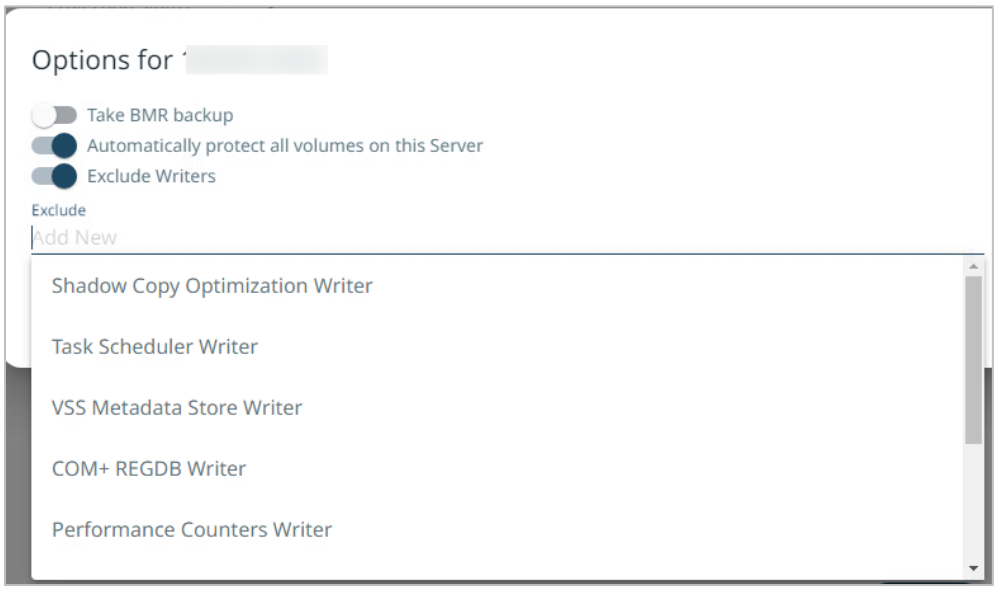
Exclude VSS Writers

Cohesity provides granular control for selecting and excluding the Windows Volume Shadow Service (VSS) writers for block-based protection. By default, all the VSS writers are involved during the backup, but these are not required to create or read the snapshot. If you want a specific VSS writer to be excluded by Cohesity (e.g - if it misbehaves during backups), it can be done in two ways. You can define an exclusion at the Protection Group level or do it for a specific source in the Protection Group via the edit source option.

Protection Group



Edit Source in Protection Group



For more information refer to the Documentation.

QoS Policy

Cohesity provides three QoS policies during the creation of a Protection Group. QoS policy affects the algorithm-based calculation for the I/O priority on the cluster. You can choose a QoS policy as per the workload requirement.

Backup HDD: Cohesity recommends using the Backup HDD QoS policy. Cohesity cluster writes all the data to the HDD drive for this Protection Group. It is suitable for workloads where higher latency is acceptable and keeps many outstanding I/Os.

Backup SSD: The Cohesity cluster writes all the data to SSDs for this Protection Group. It is suitable for small-sized Protection Groups or physical servers with many small files. Also, it can be used for workloads that require lower I/O latency and do not support many outstanding I/Os.

Backup Auto: (Applicable only for the C6K Platform) The Cohesity cluster writes data to SSDs and HDDs. Data distribution will be based on the current usage of the SSD and HDD tiers. This policy tries to achieve a similar backup performance as the **Backup SSD** policy and reduces the SSD wear-out compared to the Backup SSD policy.

Pre & Postscripts

Cohesity provides the functionality to run user-defined scripts before and after backup execution as a part of a Protection Group. You can mention the scripts in the Protection Group's Pre & Postscript section.

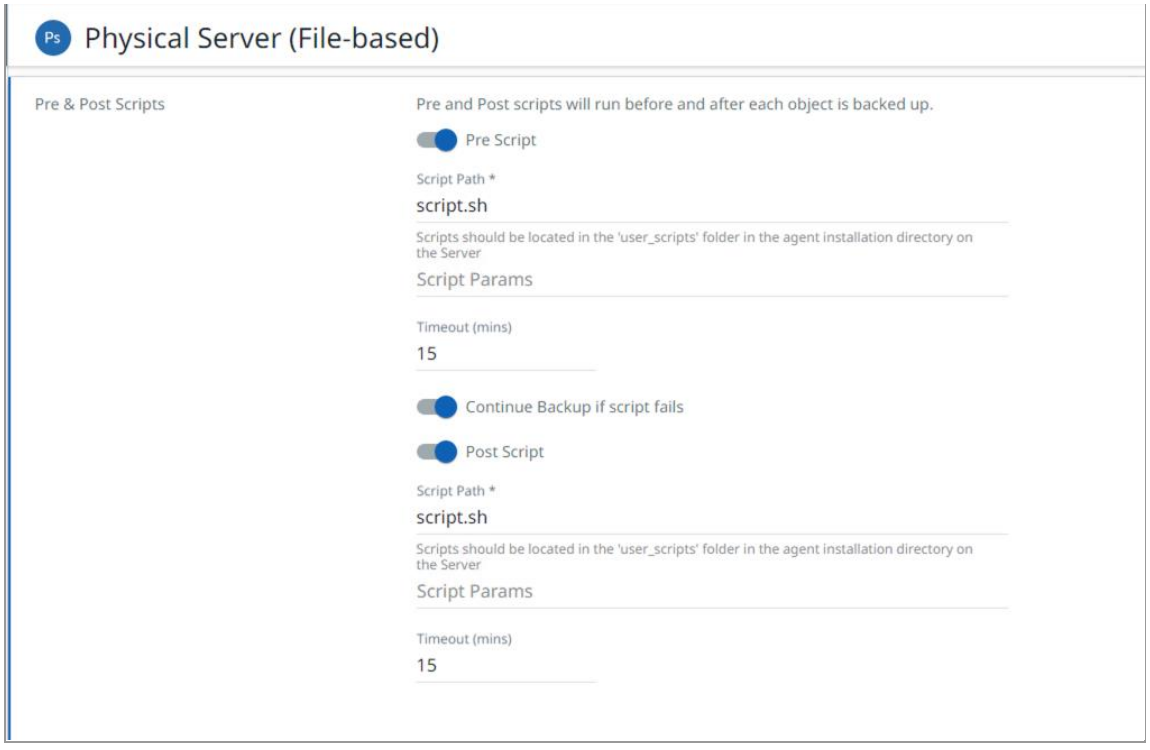
Pre & post-scripts can be used for multiple use cases. An example could be a scenario where a NAS export is backed up using a physical resource.

This would be the sequence of events:

1. Cohesity will trigger a pre-script on the proxy as part of a backup job.
2. The pre-script will connect to the NAS device and do the necessary handshaking that may include:
   a. Creating exports
   b. Creating or reusing snapshots
   c. Mounting exports for backup
3. The successful exit of the pre-script ensures that the volume designated for backup is exported.
4. The backup then takes place as usual.
5. Upon successful completion of backup, Cohesity will trigger a post-script on the proxy host.
6. The post-script will carry out the necessary actions that may include:
   a. Removal of snapshots
   b. Resetting exports
   c. Unmount operations

The following should be noted for Pre & Post-scripts in the edit or New protection group:

1. Ensure the script is executable and the user has the rights to execute the script. If you are using the *cohesityagent* user to run the backup, then the script should be owned by the *cohesityagent* user.
2. The "`shebang`" character (#!) should be mentioned in the script. For example, if a script uses bash, it should have the "`#!/usr/bin/bash`" at the start of the script.
3. The scripts should be located in the "user_scripts" directory, which is in the agent installation directory, and mentioned in the Pre & Post Scripts section.



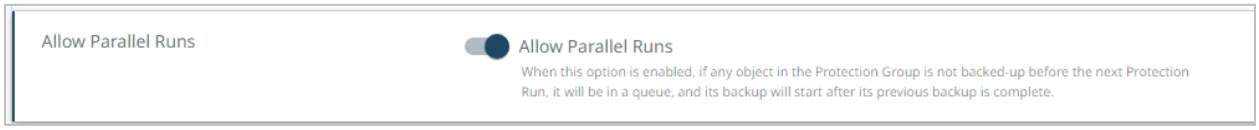
4. On Windows platforms, PowerShell, Python, or Perl scripts (or any scripts that are not executable files of the .exe form) must be embedded within a .bat script before they can be called for execution.
5. All the specified parameters for the relevant script should be passed as a single string to the script.
6. Pre and Postscripts will generate an output log file in user_script_logs folder in the installation directory of Cohesity Agent.

NOTE: Cohesity does not validate the script's content and is not responsible for the content of the script or any actions performed by the script. The user must verify the script actions before a script or executable is run. Untrusted commands and executables should not be invoked through pre- and post-scripts as they may lead to security issues.

Allow Parallel Run

Customers are facing challenges nowadays with long-running backups, particularly for specific hosts. Due to this, the backup of other sources in a Protection Group can get skipped. For example, consider a Protection Group that runs daily and has ten large physical servers; if just one physical server takes longer than 24 hours to complete, then the other nine servers will miss their next backup as the subsequent run will not start until the pending host has been backed up.

Cohesity introduced a new feature in Protection Group settings - "**Allow Parallel Runs**" - to solve the above challenges. Once enabled, if a run is executed as part of a Protection Group, new runs of the Protection Group can be parallelly started even if the current run is active. The new runs will only process sources associated with this Protection Group whose previous backups for this Protection Group are already finished.



Here's an example to illustrate the Allow Parallel Runs workflow:

Protection Group `PG1` is protecting the sources `S1, S2, S3, S4` and `S5` via backup runs `RUN1, RUN2` and `RUN3`.

1. During the first run `RUN1`, say `S1, S2`, and `S3`, complete backup before the next run, while `S4` and `S5` are still running.
2. When `RUN2` gets triggered as scheduled, it will start backing up `S1, S2`, and `S3` as scheduled but will put `S4` and `S5` in a queue or waiting mode.
3. If `S4` finishes before another run gets scheduled, then a backup of `S4` will be started by `RUN2`.
4. If `RUN3` gets scheduled and `S5` is still running as part of `RUN1`, then `RUN2` picks up available objects like `S1`, `S2` & `S3`. At the same time, `S4` is still running with `RUN2` and `S5` with `RUN1`.
5. As `RUN3` gets scheduled, any waiting objects in `RUN2` get automatically canceled and moved to a newer run like `RUN3`.
6. Also, note that if any scheduled full backup has some objects in a queue or waiting, they are never automatically canceled.

Consideration

- The same source cannot be active for two or more runs of the same job at the same time
- By default, only five parallel runs will be allowed for non-directive file backups.
- You can also trigger any new runs using the “run now” option and take advantage of parallelism.

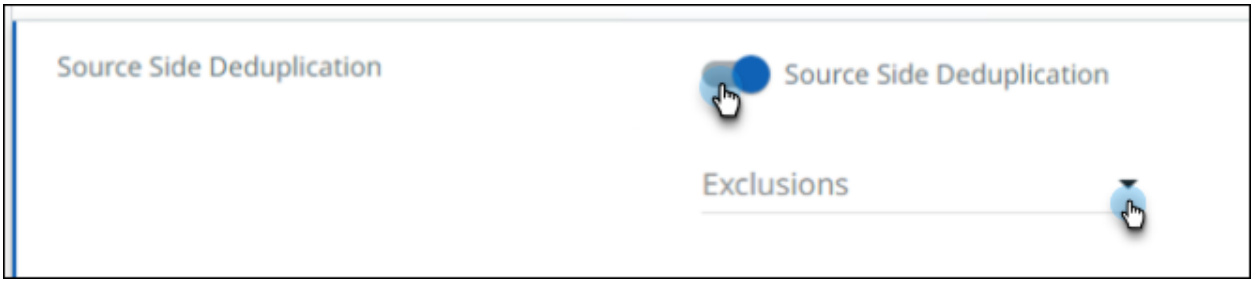
Source-Side Deduplication

Cohesity supports source-side deduplication for Windows and Linux physical servers. Source-side deduplication is a storage and network efficiency technique to aid the backup process by which only unique data is transferred over the network to a Cohesity cluster. Enabling source-side deduplication is the best solution for servers facing network bandwidth as a bottleneck.

Advantages of source-side deduplication:

- Reduces network traffic between physical servers and the Cohesity cluster.
- Leads to faster backups and better SLA adherence if data is backed up over slower networks/WAN.
- Offloading deduplication reduces CPU consumption on the Cohesity cluster and enables it to run more backup workload sessions.

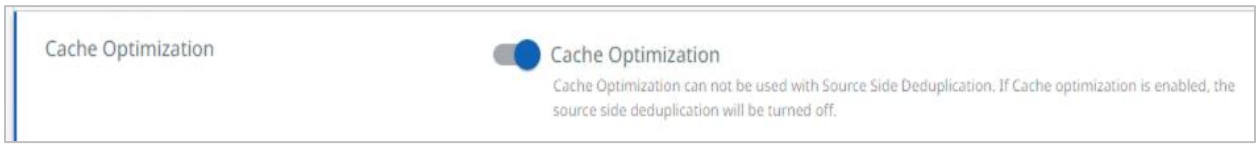
You can enable the source-side deduplication feature from the Protection Group. If your Protection Group has multiple servers and you want to exclude a specific server from the source side deduplication, you can leverage the **Exclusions** and select the servers to exclude.



Cache Optimization

Cohesity has supported source-side deduplication for Linux (x86_64) for a long time. Cache Optimization allows AIX to also leverage some source-side deduplication capabilities. The feature is a modified form of source-side deduplication that offers optimized backup performance on the Cohesity cluster. It will take a backup of only the changed blocks from the source during subsequent incremental backup. It works best when files are modified near the end of their previous content, which is true for certain workloads like Epic Cache.

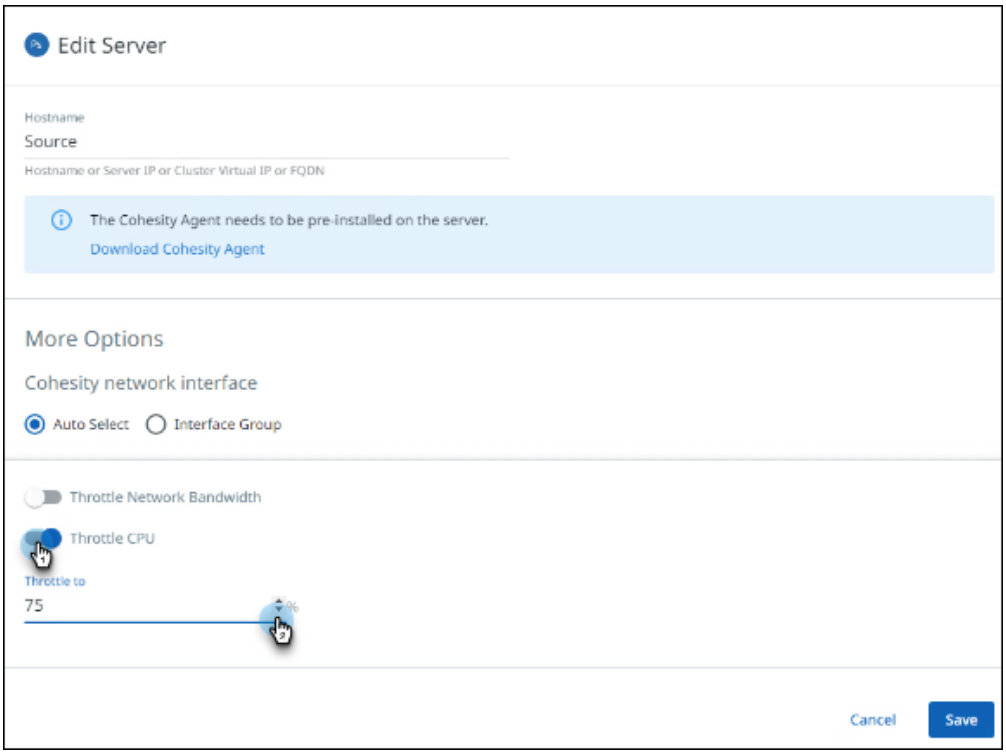
Customers benefit from reduced network utilization. The CPU utilization is also reduced in comparison to normal source-side deduplication. The feature is also available on Linux (x86_64) as an alternative to source-side deduplication. While the dedupe rate may not be as high, some customers may appreciate the reduced CPU utilization.



NOTE: Cohesity recommends enabling cache optimization only for new jobs. You should not update the older jobs with the Cache Optimization features. You can select either source-side deduplication or cache optimization for a protection job, not both.

CPU Throttling

Cohesity supports CPU throttling on Windows Server 2012 R2 and later versions of Windows. Source-side deduplication has some overhead on a physical server's CPU. During the backup process, if the CPU is over-utilized, then you can restrict the maximum CPU capacity (in percentage) of the server that source-side deduplication can utilize. You can edit the source to update the values.



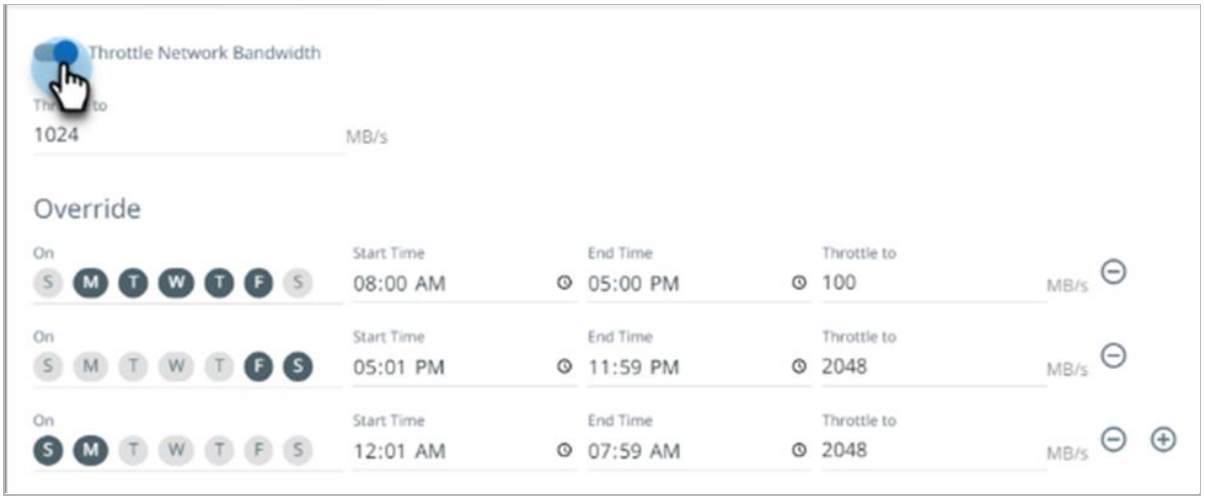
Throttle Network Bandwidth

Cohesity supports throttling the use of network bandwidth for Windows and Linux operating systems. This is supported with Windows server versions 2012 and above. For Linux, supported versions are mentioned in the supported software documentation.

You can use this feature to limit the maximum bandwidth usage from the server to the Cohesity cluster. Network throttling is measured in MB/s, with a minimum value of 1 MB/s. It allows you to schedule network throttling for backups. As the scheduler is not designed to cater to weekends wrapping around into the next week and to ensure that the start time of a block is earlier than its finish time, schedules that change the throttling limit on weekends may need to be split. Consider a network throttling requirement, as shown below:

- Throttle to 100 MB/s from Monday 8:00 AM to Friday 5:00 PM
- Throttle to 2048 MB/s from Friday 5:01 PM to Monday 07:59 AM

The period from Friday evening to Monday morning wraps around the weekend and extends into the next week. Therefore, it needs to be split so the overall schedule looks like this -



Cohesity Agent Debug Tool

Cohesity provides a debug tool for initial and basic troubleshooting or to gather/verify the information of Cohesity Agents installed on physical servers. It will allow you to manage the agent by supporting the get and update operations on Cohesity Agent.

NOTE Cohesity recommends contacting the support team for any advanced troubleshooting.

The magneto_agent_debug_tool can be used in the following ways from the Cohesity cluster -

Get agent info

```
magneto_agent_debug_tool get-info --agent_endpoints="<hostname or IP of
the physical server>”
```

Get agent logs

```
magneto_agent_debug_tool fetch-debug-files --agent_endpoints="<hostname
or IP of the physical server>”

Location of the gather logs: “/cohesity_logs/agents”
```

Get agent applied gflag details

```
magneto_agent_debug_tool get-gflag-settings --agent_endpoints="<hostname
or IP of the physical server>”
```

#Restart Cohesity Agent service from Cohesity cluster

```
magneto_agent_debug_tool restart-service --agent_endpoints=”<hostname or
IP of the physical server>”
```

Cohesity Agent Logs

During the backup of Physical sources with Cohesity Agents, avoid any issues caused by disk space getting consumed by Cohesity Agent logs. Cohesity recommends setting the log limit and retention to suitable values to avoid this. The relevant Cohesity Agent default values can be customized per the requirement and the concerned organization’s logging policy.

Table 5: Cohesity Agent Logs

Description	Default Value
Maximum log size of a log file before rolled over	30MB
Log cleanup interval in hours	1 Hour
The amount of time for which logs should be retained	30 days
The maximum amount of space allowed for storing logs	50 MB
The maximum number of core archives to retain	5
The minimum batch size for which cleanup runs	8

Cohesity provides the functionality to increase and decrease the logging level of Cohesity Agent. For example, if you want to see only the errors in the log you can set the level to 2, or vice-versa to remove the info and warning logs from the log file. Examples of logging are as follows:

- 0: Info
- 1: Warning
- 2: Error
- 3: Fatal

Cohesity Agent Port Customization

Cohesity allows you to update the communication port between Cohesity Agent on the physical server and the Cohesity cluster. By default, Cohesity Agent uses port 50051 for secure communication. In case of any special requirements, or to align with the organization's security policies, you can customize the port as required. Be sure to choose an alternative that is not used by some other application. Cohesity support can be contacted for the relevant details.

AIX Agent

To achieve enhanced backup and restore performance for the AIX server, you can verify the following aspects -

- AIX agent backup performance is good on CIO-mounted file systems. If a backup is running slower than expected, then Cohesity recommends verifying the filesystem mount options.
- If the Cohesity Java Agent on AIX takes longer than usual to start the Cohesity Agent service or to initiate a backup, then you should review stale mount points on the target servers. If there are any, then you can remove them, so the Java Agent doesn't need to process them.

General Backup Errors

[kInvalidRequest]: Cannot specify duplicated include paths in a file-based backup source, path /F/BACKUPS/file_name001.txt already exists.

Cause

- This error occurs if there are duplicate paths included in the source's file system Inclusion list.
- This error can also occur if the metadata/input file for a Directive File Backup configuration has the same file or folder path listed multiple times.

Troubleshooting

1. If this is a Physical File based backup, edit the corresponding Protection Group > Objects > Corresponding Object and validate if the same file system or directory path has a duplicate entry.
2. If this is a Directive File Backup configuration, review the metadata/input file running against the backupNow.py or backupNow.ps1 script to validate if there are any duplicate files or folder paths.

[kInvalidRequest]: The exclude path /var/tmp is not a child path of the include path /var/tmp

Cause

- This error occurs if the exclude path is not a child path of the included path.

Troubleshooting

1. Edit the corresponding **Protection Group** > **Objects** > **Entity** and validate if the Inclusion path and exclusion path have the same values because this creates a conflict when the final list of file system paths to be backed up are being evaluated.
2. Since there is a Global Exclusion list at the Protection Group level, ensure that there are no conflicting paths there as well.

   E.g.: if both the Object Inclusion and Global Exclusion path have /var/tmp, this error occurs.

[kInvalidRequest]: Either one of, an explicit list of backup paths or, the file containing a list of backup paths must be provided for the backup source.

Cause

- This error occurs if a Directive File Backup configuration is set up but does not specify the metadata/input file when running the backup scripts or specifies an invalid/non-existent file name.

Troubleshooting

1. Validate it when the backupNow.py or backupNow.ps1 scripts are being triggered, provided there is a valid metadata/input file.
2. Check if the metadata/input file exists, or if there's a typo in the file path.

[kInvalidRequest]: Include path /D could not be found.

Cause

- Failing a protection job when the inclusion path cannot be found is the expected behavior.
- During a Cohesity Protection Group run, the cluster sends a request to the source server to retrieve a list of available volumes. The job fails when a path that has been explicitly included is missing.

Troubleshooting

1. Investigate why the file system or volume path that is part of the Inclusion list is not actually present on the source.
2. Check If this file system or volume path does not exist anymore on the source and remove it from the inclusion list.

Host Agent Communication Errors

Error(s)

- [kTransportError]: Cohesity service on host {source} cannot be reached.
- [kTransportError]: Connection failure to server.domain.com during the call GetAgentInfo.
- [kTransportError]: Cohesity service on host {source} cannot be reached.
- [kTransportError]: RPC call GetAgentInfo timed out.

Cause

- These errors occur if there is any communication issue with the Cohesity Agent on the host.
- These errors can have multiple root causes that require further investigation to troubleshoot the root cause.

Troubleshooting

1. Confirm the host is powered on.
2. Confirm Cohesity Agent on the host is running -
   a. Windows (CLI) - To check the agent status, use the following command:

```
C:\Users\Administrator>sc query cohesityagent
SERVICE_NAME     :     cohesityagent
TYPE             :     10 WIN32_OWN_PROCESS
STATE            :     4 RUNNING(STOPPABLE, NOT_PAUSABLE,
IGNORES_SHUTDOWN)
WIN32_EXIT_CODE  : 0 (0x0)
SERVICE_EXIT_CODE : 0  (0x0)
CHECKPOINT       : 0x0
WAIT_HINT        : 0x0
```

   b. Check the Cohesity Agent status from the Windows services GUI tab.
      - Confirm whether it is turned on and set to "Automatic", NOT "Manual".
   c. Linux - To check the agent status, use the following commands:
      - `ps -ef |grep cohesityagent`
      - `ps aux | grep -i cohesityagent`
3. Verify connectivity between the Cohesity cluster and the target server -
   - Access the Cohesity bash shell. From the shell, attempt to ping the server. If pings fail, investigate a network issue.
   - Confirm port 50051 is open.

```
sudo nmap -p 50001 -sT host_name
```

   - Confirm agent communication from the Cohesity cluster to the agent host.

```
magneto_agent_debug_tool get-info --agent_endpoints=Host IP
Address
```

   - Registering a source with a short name may allow nslookup to resolve the address from the Cohesity nodes, irrespective of registration performed with a FQDN or IP address.
4. Restart the Cohesity Agent service on the host.
   - On a Windows server, click **Start** and launch services.msc. Locate and restart the Cohesity Agent service.

[kFsError]: Cohesity could not visit any entity successfully. Please verify that the source is reachable from the cluster and that Cohesity has the right access credentials: [kWindowsSystemError]: Failed to get file attributes.

Cause

- This is typically a Windows permissions issue that requires further investigation to troubleshoot the root cause.
- The credentials to access the filesystem in question could have changed.
- The domain account being used to register the host with the cluster is not in the domain(s) that the cluster has been joined to.
- The Windows server is not in a domain that is trusted by the domain(s) the cluster is joined to.
- The Windows server is in a domain that is denied in the cluster's Active Directory configuration.

Troubleshooting

- Confirm the credentials have the correct permissions and access to the host.

[kNonExistent]: [kNotFound]: File::Initialize() failed in CreateFile of e:\Cohesity_Backup_List\Cohesity_filelist.txt : 0x2 : The system cannot find the file specified"

Cause

- This error can occur if the metadata / input file that is needed in a Directive File Backup is non-existent on the source server.

Troubleshooting

- Confirm if the metadata / input file is present in the mentioned path as specified as an argument in the backupNow.ps1 or backupNow.py scripts.

Appendix A: Troubleshoot WinRM Issue

WinRM is a prerequisite for deploying Windows agents from the Cohesity dashboard. The agent deployment may fail with the following error -

Error: Failed to connect through WinRM: Failed to connect windows machine: http response error: 401 - invalid content type

In such a scenario, you can follow the subsequent steps to fix the WinRM issue.

1. As we access Windows hosts through WinRM via the HTTP mode, enable HTTP access by using the below commands:

```
winrm set winrm/config/service/Auth '@{Basic="true"}'
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="1024"}'
```

2. Allow WinRM HTTP and WinRM HTTPS in the Windows firewall.
3. Restart the Windows machine.
4. If the machine is in a domain, try to unjoin and rejoin. Then restart the host.
5. Make sure that the Windows machine name is unique, otherwise, WinRM will not work properly.

Appendix B: CSV Preparation

Before uploading the CSV file for deployment, make sure you follow all these points -

1. The Sample File downloaded from the Cluster UI is in the “txt” format. After entering the information, change the file format to the CSV format and then upload.
2. Each row of the CSV should have 13 columns.
3. The CSV deployment method supports mixed operating system deployment. You can enter Linux and Windows operating system servers in one CSV.

IP/Hostname	OS Type	Server Username	Server Password	Install Location
10.0.0.1	linux	admin	test_password	/var/agent_location/
10.0.0.2	windows	admin	test_password	C:\\Program Files\

4. Remove the first row, which has the header information for the category of all the columns, as Cohesity will consider that row as a value for deployment.
5. Reverification of the CSV file can be done by opening it in a text editor. Check that every column should end in a Comma (,) even if blank. Each line should have 13 commas in a line.

```
IP/Hostname,OS Type,Server Username,Server Password,Install Location,Volume CBT,File CBT,Windows Serv
user,Linux service group,Linux create service user
10.0.0.1,linux,admin,test_password,/var/agent_location/,,,,,,oracle,oracle,FALSE,
10.0.0.2,windows,admin,test_password,C:\\Program Files\,,,service_admin,service_password,,,,
```

6. The final form of the CSV file resembles this:

A	B	C	D	E	F	G	H	I	J	K	L	M
10.0.0.1	linux	admin	test_password	/var/agent_location/						oracle	oracle	FALSE
10.0.0.2	windows	admin	test_password	C:\\Program Files\			service_admin	service_password				

7. If you face this issue *Error: “Invalid number of columns in one or more rows.”,* it means that the CSV file you are uploading has a missing entry. Verify that the file meets all the above guidelines.

Your Feedback

Was this document helpful? Send us your feedback!

About the Authors

Punit Gupta is a Staff Technical Solutions Engineer at Cohesity. In his role, Punit focuses on data protection, physical agents, SmartFiles, Multitenancy and Cohesity storage as a backup target.

Document Version History

VERSION	DATE	DOCUMENT HISTORY
3.0	Aug 2024	Content Updated for 7.1.2 version
2.0	Oct 2023	First Public Release
1.0	June 2023	Internal Release

ABOUT COHESITY

Cohesity is a leader in AI-powered data security and management. Aided by an extensive ecosystem of partners, Cohesity makes it easier to protect, manage, and get value from data – across the data center, edge, and cloud. Cohesity helps organizations defend against cybersecurity threats with comprehensive data security and management capabilities, including immutable backup snapshots, AI-based threat detection, monitoring for malicious behavior, and rapid recovery at scale. Cohesity solutions are delivered as a service, self-managed, or provided by a Cohesity-powered partner. Cohesity is headquartered in San Jose, CA, and is trusted by the world's largest enterprises, including six of the Fortune 10 and 44 of the Fortune 100.

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