# Technical Solution Guide: Implementing Resilio Active Everywhere as a Global File System (GFS)

## Introduction

In today's enterprise landscape, the demand for a robust, scalable, and efficient Global File System (GFS) is more critical than ever. Organizations are managing distributed teams, handling massive datasets, and requiring real-time collaboration across multiple geographic locations.

Resilio Active Everywhere is a powerful GFS solution that is a software layer on top of existing storage and infrastructure.  As such, it doesn't require data migration and can be deployed much faster than competing solutions.

Leveraging a unique P2P approach and sophisticated caching capabilities, Resilio delivers high-performance file synchronization and access.

Resilio offers a distinct approach tailored to the technical demands of IT professionals and DevOps teams. The solution is faster, more scalable, more flexible, and more automation-friendly than competing solutions.

This guide is intended for IT professionals deploying Resilio Active Everywhere for evaluation purposes. It explains how to implement a basic setup with a single Primary Storage and Caching Gateway, and then expand it with additional instances for improved resiliency and performance.

The document also covers some of the key features of Resilio Active Everywhere, such as Distributed File Locking, Revision History and caching capabilities.

## Value Proposition

Resilio Active Everywhere is built upon a peer-to-peer (P2P) architecture and as a layer on top of existing storage, diverging from traditional cloud-based or centralized file systems. This architecture enables:

- Efficient Data Transfer: Utilizing the P2P, UDP-based protocol for fast, reliable file synchronization.
- Scalability: Seamless scaling from millions to hundreds of millions of files and from a few devices to tens of thousands without bottlenecks.
- Resilience: Eliminating single points of failure inherent in centralized systems.
- Fast deployment: Lift and shift is optional. Resilio runs on top of existing storage, file systems, and buckets.

- VPN-less access: Simplifying secure access to shared data.

Key Features:

- Server Caching Capabilities: Intelligent caching mechanisms reduce data latency and bandwidth usage.
- Distributed File Locking: Ensures data consistency and prevents conflicts in collaborative environments.
- Real-Time Synchronization: Immediate propagation of file changes across all nodes.
- WAN Optimization: Enhanced data transfer over wide-area networks.
- File streaming: Instantaneous access to large files by only transferring the file chunks that are needed.

## Competitive Landscape: Resilio Active Everywhere vs. Others

### Architectural Comparison

| **Feature** | **Resilio Active Everywhere** | **Competition (Panzura, CTERA, Azure Files, etc.)** |
|---|---|---|
| Architecture | P2P distributed architecture | Hub-and-spoke: Cloud-based hub with branch caches |
| Data Storage | Flexible: any host, any NAS, any cloud provider | Proprietary Cloud Storage |
| Latency Optimization | P2P direct transfers using a UDP-based protocol | Transfers through the cloud hub over TCP (latency- and distance-sensitive) |
| Distributed File Locking | Fast, P2P lock propagation, sub 5 seconds | Centralized locking servers, slow to respond |
| Storage Format | Native: Files and objects are accessible without the Resilio layer | Proprietary: The data is not accessible without a third-party solution, and, for example, is not available for cloud-native processing. |
| Egress Costs | Lower costs: Cloud usage is optional. If used, direct P2P transfers reduce egress. | Higher costs: Every change is downloaded from the cloud to multiple locations. |

## Technical Architecture of Resilio Active Everywhere

#### GFS as a Software Layer

Resilio Active Everywhere is a software-only solution:

- Uses existing storage: No need for data migration.

- No vendor lock-in: Data is stored in native format.
- Flexible and scalable: Scale your deployment as your business grows. Add Caching Gateways in strategic locations to optimize data access.

### Peer-to-Peer Core

At the heart of Resilio Active Everywhere lies its P2P protocol, which ensures:

- Parallel Data Transfers: Files are split into blocks and transferred simultaneously across multiple peers.
- Efficient Bandwidth Utilization: Dynamic bandwidth adjustment based on network conditions.
- Resilience: Automatic failover and resumption of transfers.

### Server Caching Mechanisms

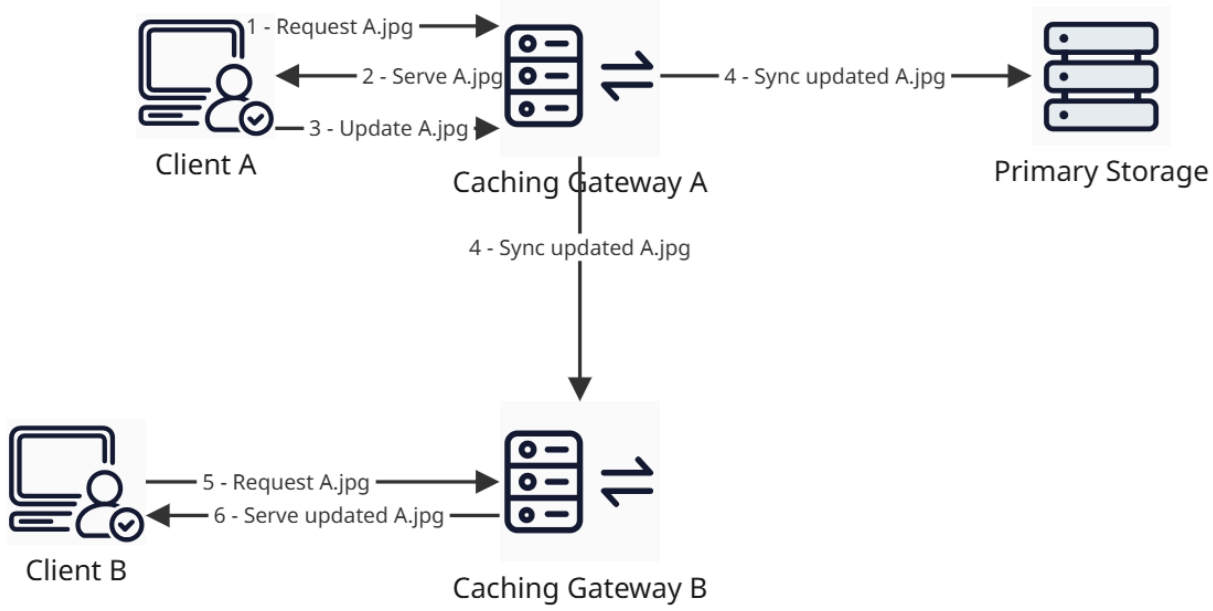
Resilio's server caching capabilities optimize data access:

- Local Caching Gateways: Deployed at each site to serve frequent file requests rapidly.
- Differential Synchronization: Only changed portions of files are synchronized, reducing transfer sizes.
- Cache Validation: Intelligent validation to ensure data consistency across the network.

### VPN-less Access Mechanism

Built-in end-to-end encryption enables secure access to shared data without a VPN.

Cache Workflow Diagram:



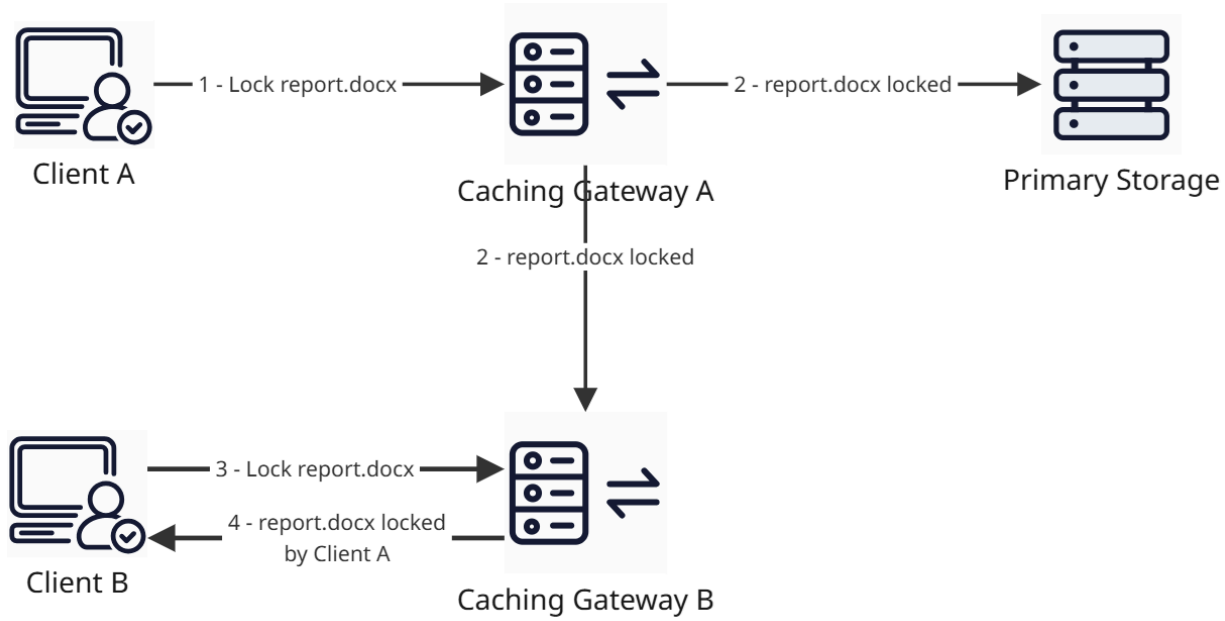
- Step 1: Client A requests a file.
- Step 2: If the file is available, the local Caching Gateway serves it directly.
- Step 3: User A updates the file.
- Step 4: The file update is synchronized between the other Caching Gateway and the Primary Storage
- Step 5: Client B requests this file.
- Step 6: Client B retrieves the updated file from the Caching Gateway without accessing central storage.

## Distributed File Locking

Resilio's file-locking feature enhances collaboration and efficiency in distributed environments:

- Local Locking Information: Lock information is maintained locally, ensuring immediate availability and reducing latency.
- In-Memory Index: Maintains an in-memory index of the file system to ensure fast lock propagation.
- Offline Work Support: Desktops and laptops can cache files locally, lock them for editing, and go offline while maintaining the lock.

Lock Workflow Diagram:




Step 1: Client A asks to lock a file.

- Step 2: Caching GW A checks its local file locking database, locks the file if available and notifies the central locking server running on the Primary Storage as well as the local file locking service on the Caching Gateway B.
- Step 3: Client B asks to lock the same file.

- Step 4: Caching Gateway B checks its local file locking database and notifies Client B that the file is already locked.

#### Security and Encryption

- End-to-End Encryption: All data transfers are encrypted end-to-end using TLS and AES-256.
- File ACLs Replication: Granular permissions to control user access at the file and folder level.
- Authentication: Integration with user directories for admin authentication.

# Implementation Guide

## System Requirements

#### Hardware and Operating Systems

- Operating Systems: Supports Windows, Linux, and macOS.
- Hardware: Commodity servers for Cache Servers; no specialized hardware required. The system can be fully virtualized.
- For more information, see System Requirements - Resilio Help Center

#### Network and Firewalls

- For a list of ports and protocols used by Resilio Active Everywhere, see Ports Protocols Required by Resilio Active Everywhere - Resilio Help Center.
- A deployment using a proxy in the DMZ is also supported. For details, see Installing and configuring proxy server - Resilio Help Center.

## Deployment Steps Overview

1. Prepare Infrastructure:
    - Identify optimal locations for Caching Gateway deployment.
    - Ensure network connectivity between nodes.
2. Install the Resilio Management Console and apply a license.
3. Set Up Admin Access Controls:
    - Create user accounts and user groups for administrators allowed to access the Management Console and manage Jobs, Profiles, Policies, etc.
    - Integrate with existing directory services (Entra ID, Okta).
    - Assign permissions and roles to users and groups.
4. *(Optional)* Configure Cloud Storage(s):
    - Configure Storage Connectors for any object storage bucket you plan to use.

5. Install Resilio Agents:
    - Deploy Agents on all servers and end-user devices.
6. Create Jobs:
    - Define synchronization jobs specifying primary storages, caching gateways (optional), and end-user devices (optional).
7. Test Synchronization:
    - Perform initial synchronization testing with test data.
    - Monitor transfer speeds, cache hits/misses, and data integrity.
8. Optimize Using Caching Policies:
    - Tweak your File Policies to optimize the caching performance.

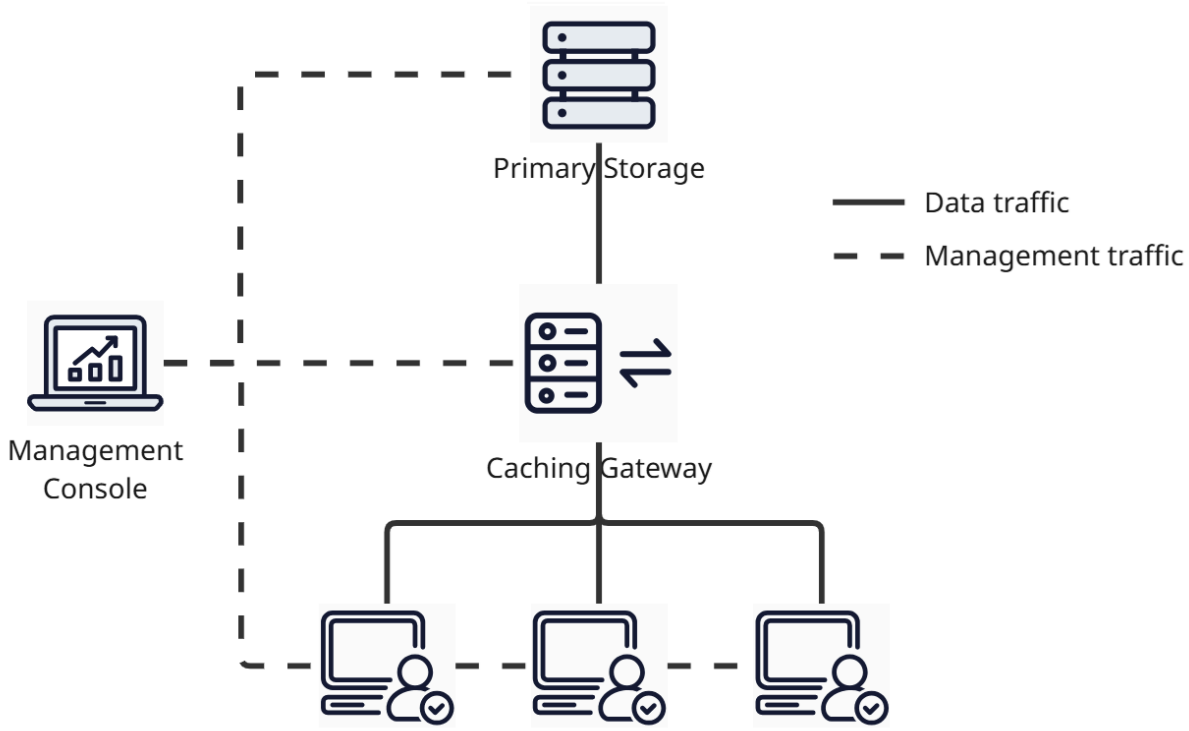
## Planning and Architecture Overview

In this section, we provide a planning and architecture overview for deploying Resilio Active Everywhere. Our focus is on creating a high-level design that ensures scalability, reliability, security, and performance. We will outline the key architectural components and their interactions, supported by logical diagrams. These visual aids will illustrate the overall structure of the system, enabling a clear understanding of how the various elements work together to support your production workloads. The first section of the reference architecture will give you a quick look at a minimal deployment before we start adding elements and discussing different deployment details.

### High-Level Design for a Minimally Viable Architecture

#### Overview

The diagram below shows a high-level design for a minimally viable deployment. After explaining the core elements, we will gradually expand this deployment to provide additional functionality.




Configuration

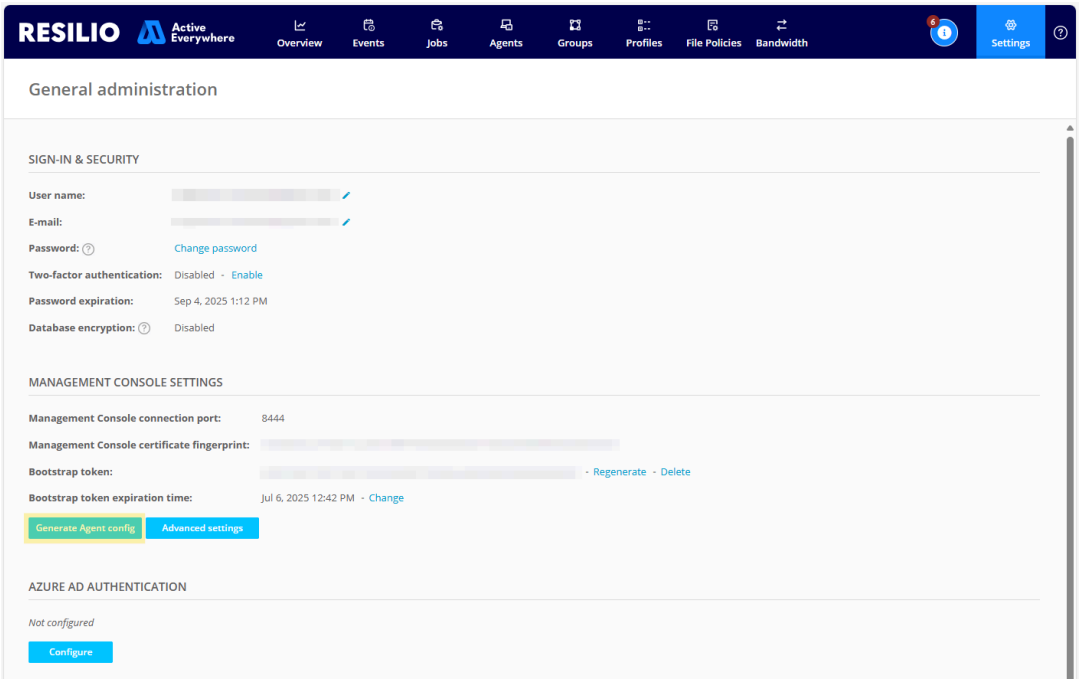
*Installing Agents*

**Generate Configuration File**

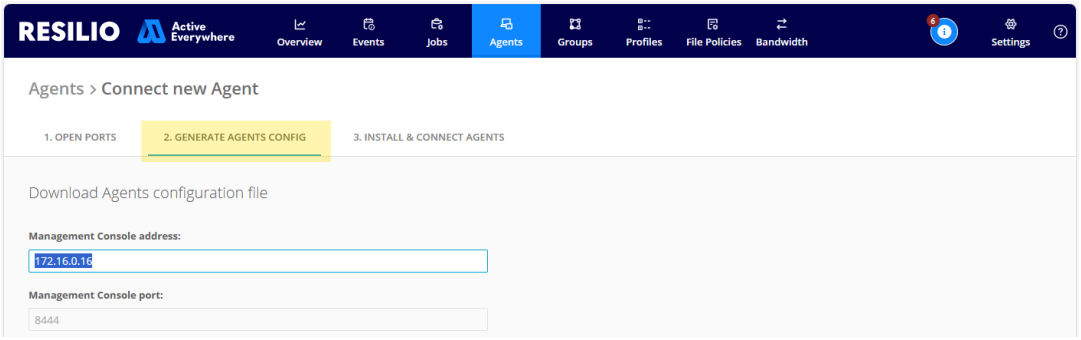
The configuration file contains a set of parameters that enable linking a Resilio Agent instance to the Management Console. While there are options to install and provision Agents without the configuration file, it's recommended that you use the configuration file to connect Agents to the Management Console.

To generate a configuration file:

1. Log in to the Management Console.
2. From the **Settings** menu, select **General** and in the **MANAGEMENT CONSOLE SETTINGS** section, click **Generate Agent config**.




or navigate to **Agents** , then click **+ CONNECT NEW AGENT** and select **GENERATE AGENTS CONFIG**.




3. Provide the following:
    - **Management Console address** - The Management Console's IP address or hostname.
    - **Management Console port** - The port number on which the Management Console listens for incoming connections.
      **Note**: Management Console's incoming connections port number is read only. It can be changed by editing the configuration file.
    - **Bootstrap token** - The bootstrap token authenticates the connection between Resilio Agents and the Management Console.
    - **Default storage path** - The default storage path for the jobs.
    - **Enable Agent UI** - Enables user interface for Resilio Agents running on Microsoft Windows and macOS.
    - **Add Agent tags** - Tags enable automated assignment of Agents to groups.

4. Click **Save**/**Download**.

**Install Agent - Windows**

To install Resilio Agent on Windows:

1. Download the Agent installer.
2. Copy the Agent configuration file to the folder where the installer is located.
3. Launch the installer.
4. On the welcome page, click **Next**.
5. (*Optional*) Customize your installation, then click **Next**.
   - **Resilio Connect Agent Service** - The Resilio Agent service that enables participating in different transfer jobs.
   - **Firewall exception** - Rules in Windows Firewall allowing Resilio Active Everywhere to establish network connections.
   - **File Locking driver** - OS level driver that enables the file locking feature. **Important**: Installing the File Locking driver later, after the Resilio Agent has been installed initially, is difficult. We highly recommend installing the locking driver with the Resilio Agent if you plan on using file locking.
6. Enter the destination folder, then click **Next**.
7. Click **Install** to begin the installation.

**Install Agent - macOS**

**Note**: We will use a manual method to install the Agent. It is possible to package the configuration file with the Agent application.

To install Resilio Agent on macOS:

1. Download the Agent installer.
2. Open the installer image, then drag and drop the Resilio Agent on to the **Applications** folder.
3. Open the Resilio Agent application.
4. On the **Configure connection with Management Console** screen, select **Config file**.
5. Click **Choose File**, then locate the configuration file, and click **Ok**.

**Install Agent - Linux**

**Tarball**
To install Resilio Agent from a tarball archive:

1. Download, then extract the Resilio Agent tarball archive:
   `tar zxf resilio-connect-agent_x64.tar.gz -C /home/vboxuser/apps/resilio-agent`
2. Start the Resilio Agent with specifying the location of the configuration file:
   `./rslagent --config /home/vboxuser/Downloads/sync.conf`
3. In the `/lib/systemd/system/` folder, create a unit file `resilio-agent.service` with the following content:

```
[Unit]
Description=Resilio Agent service
Documentation=https://connect.resilio.com
After=network.target network-online.target

[Service]
Type=forking
UMask=0002
Restart=on-failure
PermissionsStartOnly=true

LimitNOFILE=16384

User=vboxuser
Group=vboxuser
Environment="AGENT_USER=vboxuser"
Environment="AGENT_GROUP=vboxuser"

Environment="AGENT_LIB_DIR=/var/lib/resilio-agent"
Environment="AGENT_CONF_DIR=/etc/resilio-agent"

PIDFile=/var/lib/resilio-agent/sync.pid

ExecStartPre=/bin/mkdir -p ${AGENT_LIB_DIR}
ExecStartPre=/bin/chown -R ${AGENT_USER}:${AGENT_GROUP}
${AGENT_LIB_DIR}
ExecStart=/home/vboxuser/apps/resilio-agent/rslagent --config
${AGENT_CONF_DIR}/sync.conf --storage ${AGENT_LIB_DIR}
ExecStartPost=/bin/sleep 1

[Install]
WantedBy=multi-user.target
```

4. Run the `systemctl daemon-reload` command to reload unit files.
5. Run `systemctl enable resilio-agent` to enable autostart for the Resilio Agent service.
6. Run `systemctl start resilio-agent`.

**DEB package**
To install Resilio Agent from a DEB package:

1. Download the .deb package, then run the following command:
   `sudo dpkg -i resilio-agent_<version>_<architecture>.deb`
2. Enable automatic startup of the Resilio Agent service:
   `sudo systemctl enable resilio-agent`
3. Start the Resilio Agent service:
   `sudo systemctl start resilio-agent`

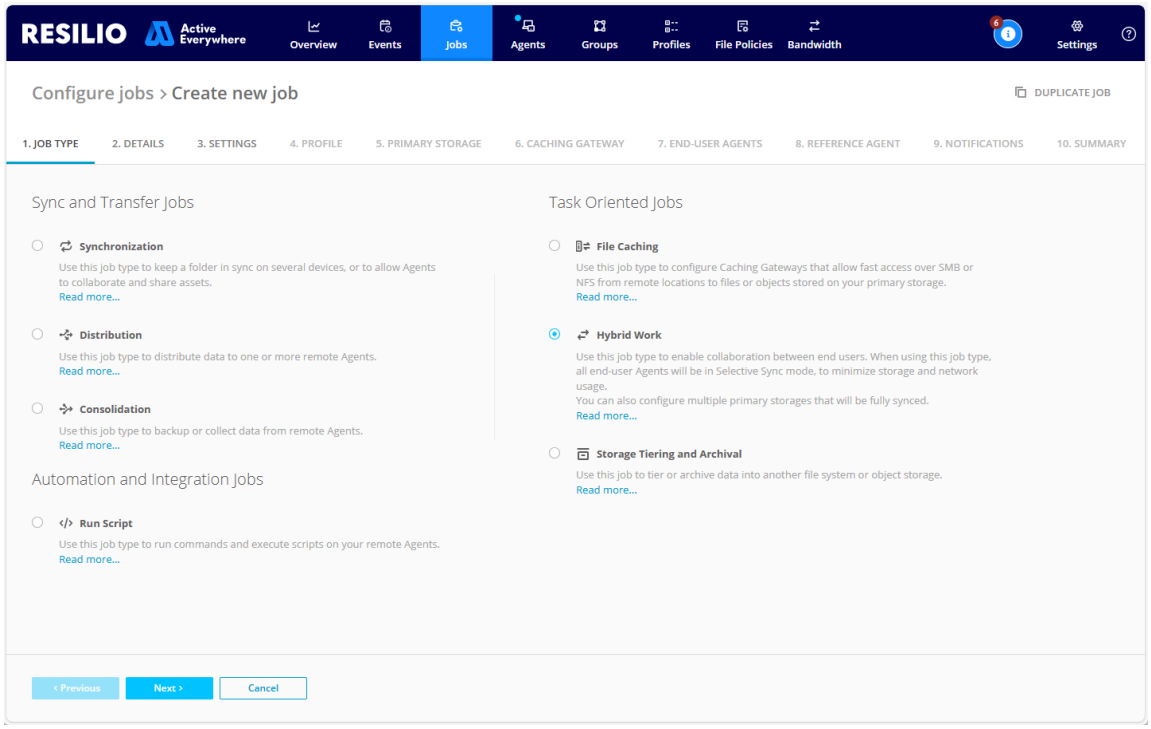
**RPM package**
To install Resilio Agent from an RPM:

1. Download the `sync.conf` configuration file and place it in the `/etc/resilio-agent` folder.
2. Download the .rpm package, then run the following command:
   `sudo rpm -Uhv resilio-agent_<version>_<architecture>.rpm`
   Enable automatic startup of the Resilio Agent service:
   `sudo systemctl enable resilio-agent`
3. Start the Resilio Agent service:
   `sudo systemctl start resilio-agent`

*Creating a Job*

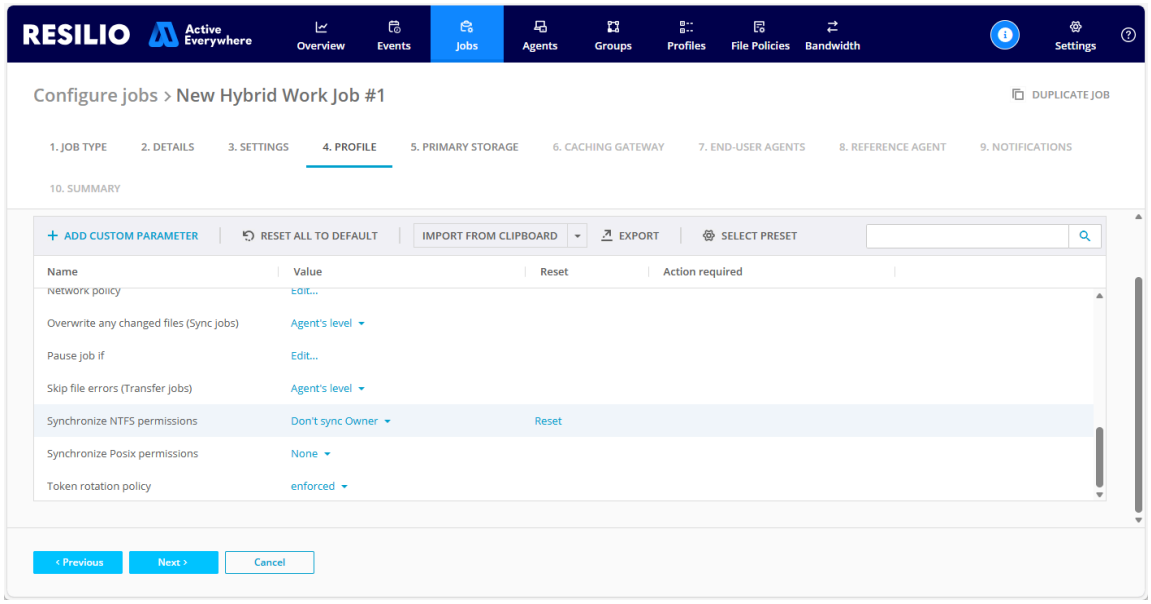
1. Select **Jobs** > **Configure Jobs** and click **+ CREATE NEW JOB**.

2. Select **Hybrid Work** and click **Next**.




3. In the **Details** tab, name your Job and click **Next**.
4. In the **Profile** tab, adjust job settings to enable permissions synchronization:

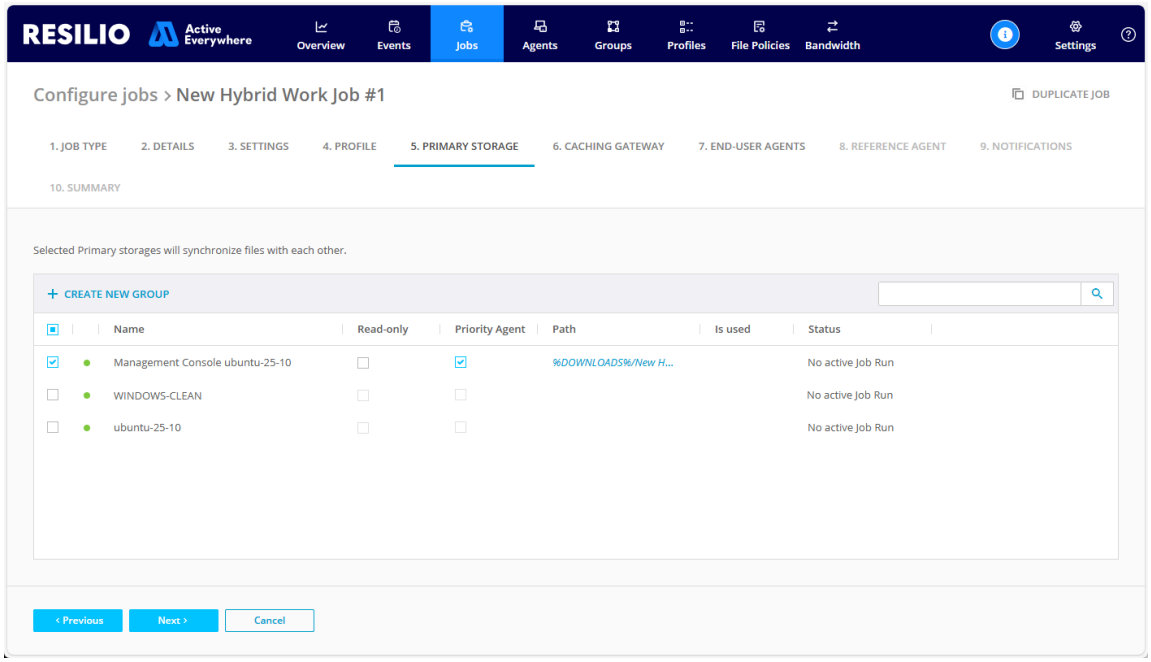
   a. From the **Job settings** drop-down list, select **Unique for this job**.
   b. Locate the **Synchronize NTFS permissions** parameter and set its value to **Don't sync Owner**.
   c. Click **Next**.




5. In the **Primary Storage** tab, configure your Primary Storage hosts.

   The Primary Storage stores the golden copy of your data. The data can reside on Direct Attached Storage (block storage), Network Attached Storage (NAS) mounted over SMB or NFS, or Object Storage accessed over S3-like API. Azure Files is also supported.

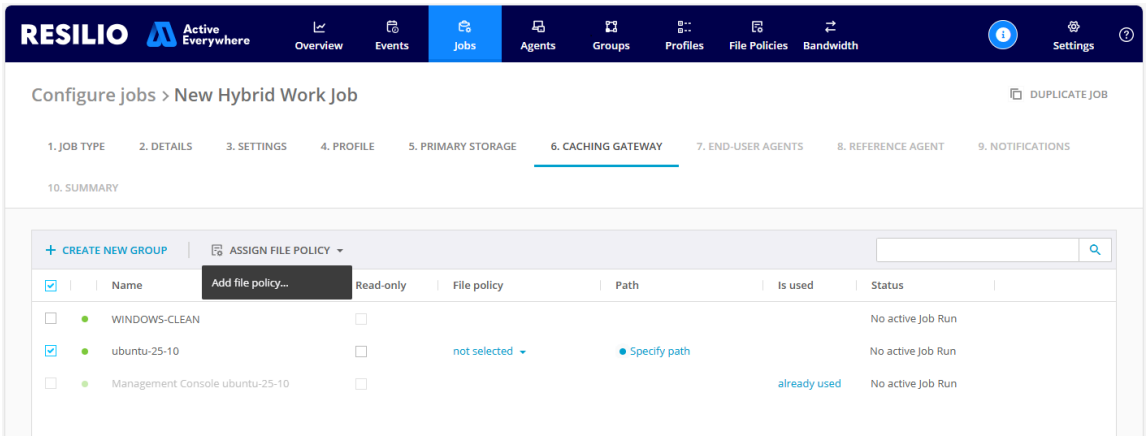
   Select your Primary Storage instance and specify the path where the data for this Job is stored.



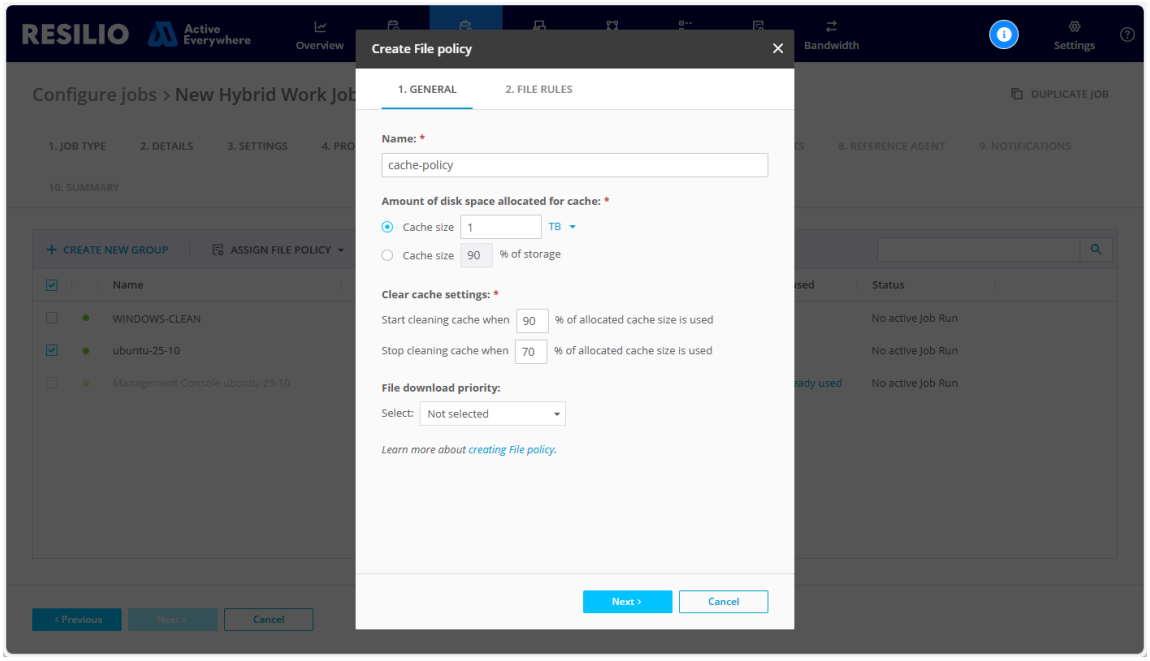
6. Configure your Caching Gateways.

   The Caching Gateways should be deployed in close proximity to users and applications that consume the data. Files will be delivered to the Caching Gateway using Resilio's WAN-optimized protocol. The Caching Gateway exposes the files over SMB or NFS. To use SMB, deploy Windows Servers. To use NFS, deploy Linux servers.

   a. Configure a File Policy.
   A File Policy is required to complete the configuration of a Caching Gateway. The File Policy defines the cache max size, as well as pre-caching and pinning rules. For now, let's create a blank policy.  We will edit it later.

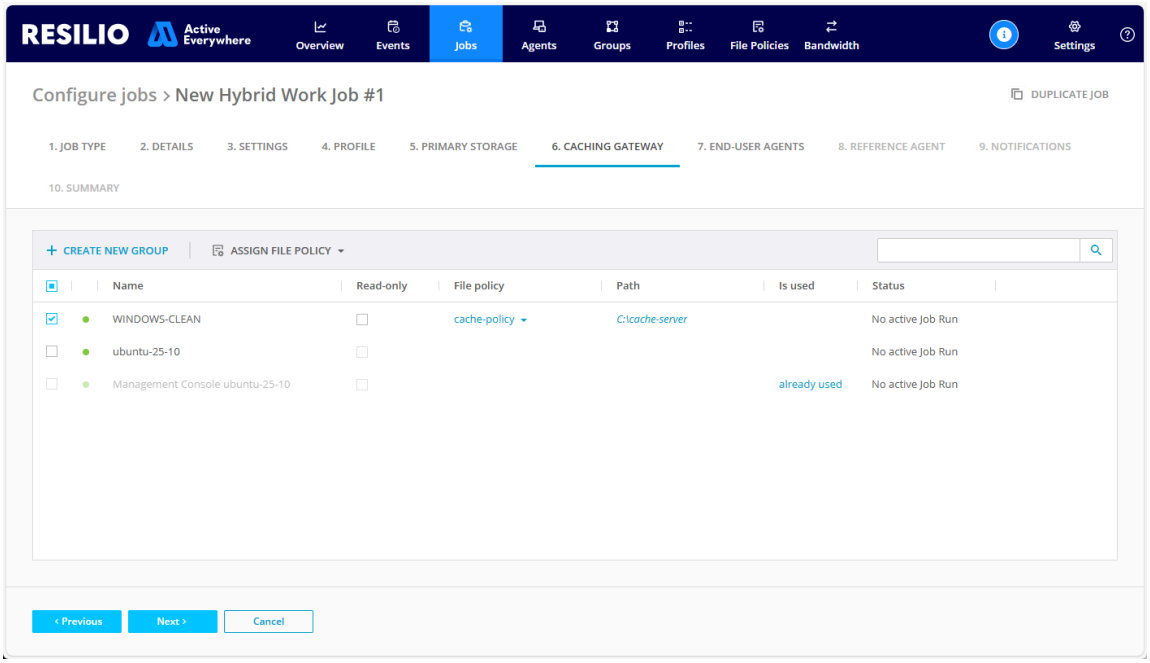



   Name your policy and set the storage space allocated to caching data.  Later we will discuss cache size considerations and ways to measure the cache efficiency.




Leave the **File Rules** settings empty for now, then click **Save**.

b. Select a Caching Gateway.
Select the device that will be your Caching Gateway, assign a previously created **File policy** to it, and pick a folder for the cache location. Keep in mind that this folder will be shared over SMB or NFS. You can pick an existing share for the cache.



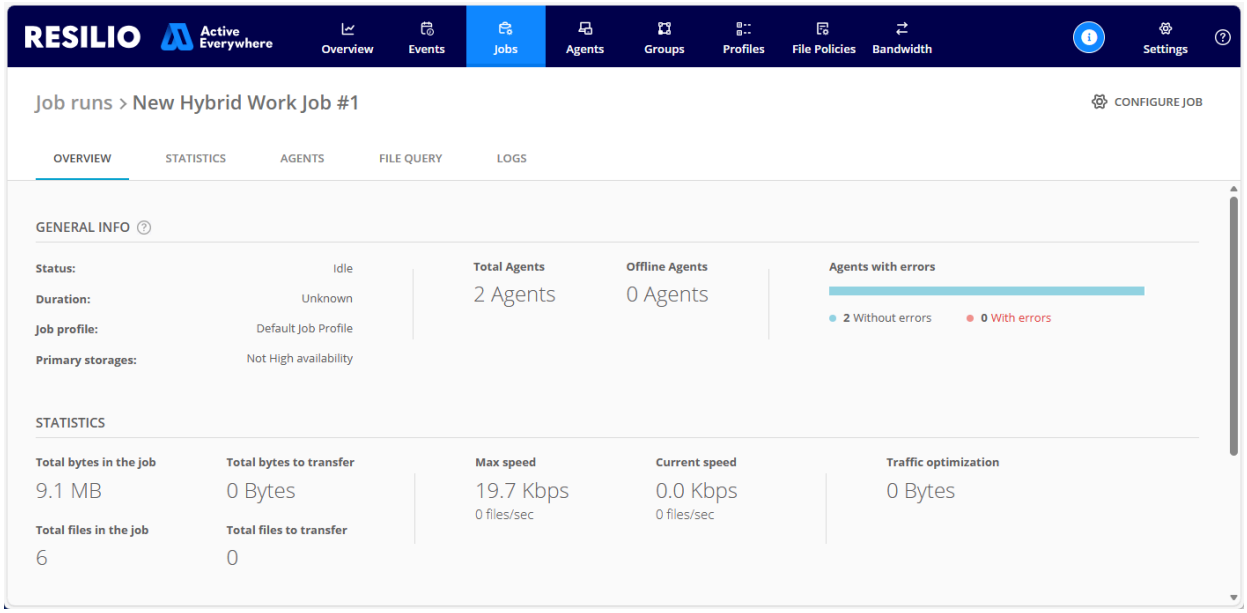
7. Skip the remaining configuration steps and save the Job.

8. Configure the Caching Gateway host.
   If using Windows, configure the folder to be shared.
   If using Linux, follow this guide to configure the NFS share - [Configuring Linux OS cache servers – Resilio Active Everywhere]

## Review

We now completed the configuration of the minimally viable architecture.
Navigate to **Jobs** > **Job Runs**, then click the Job Run that corresponds to the Hybrid Work Job that you've created.

We have two Agents: one Primary Storage and one Caching Gateway.
Very few files if at all were transferred.

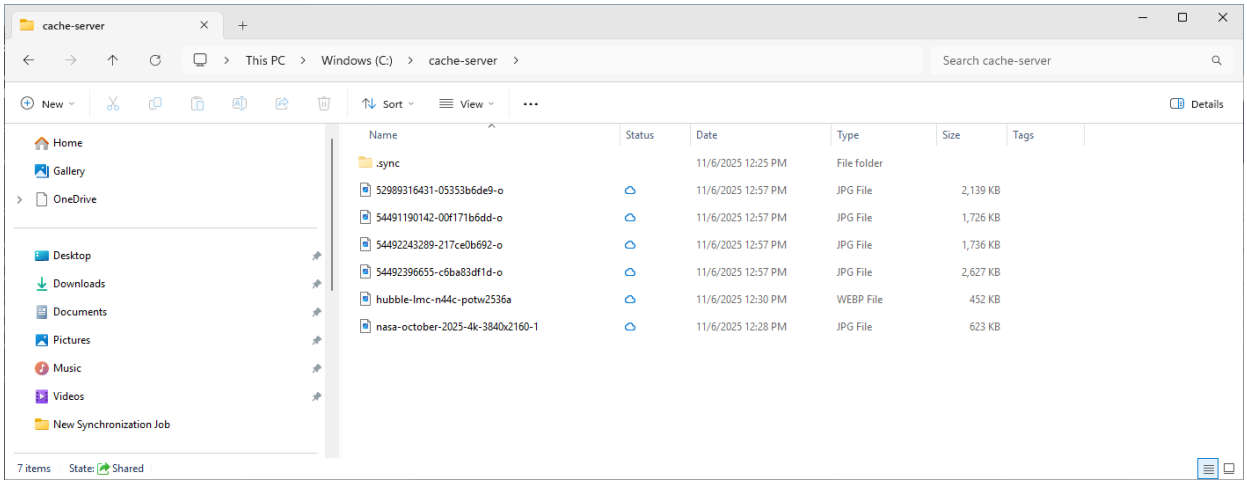



## Testing

### *The Caching Gateway View*

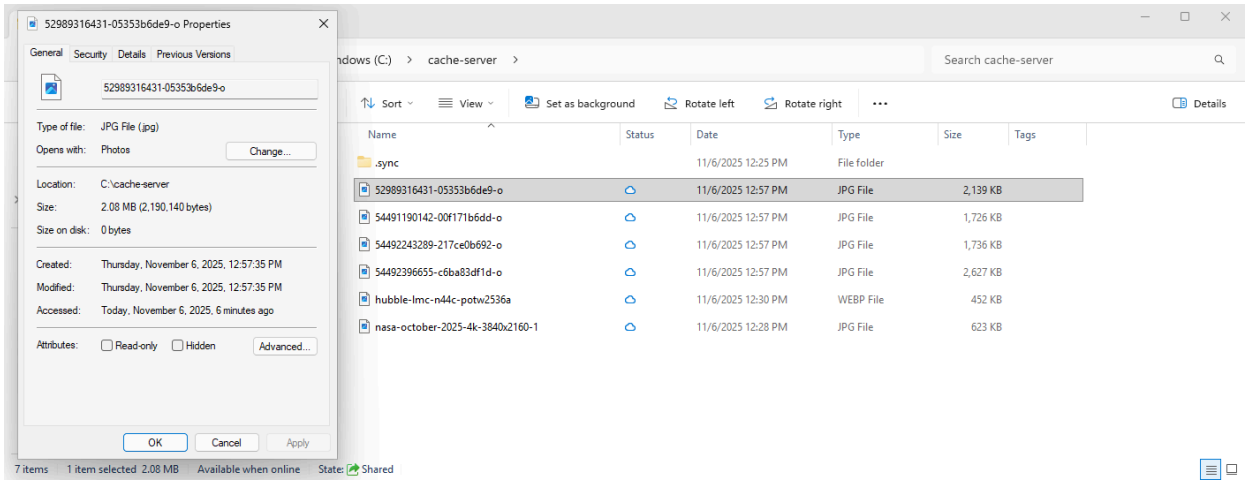
The Caching Gateway should show placeholders for all of the files in the Job folder on the Primary Storage.

This is what you can expect to see on a Windows machine:



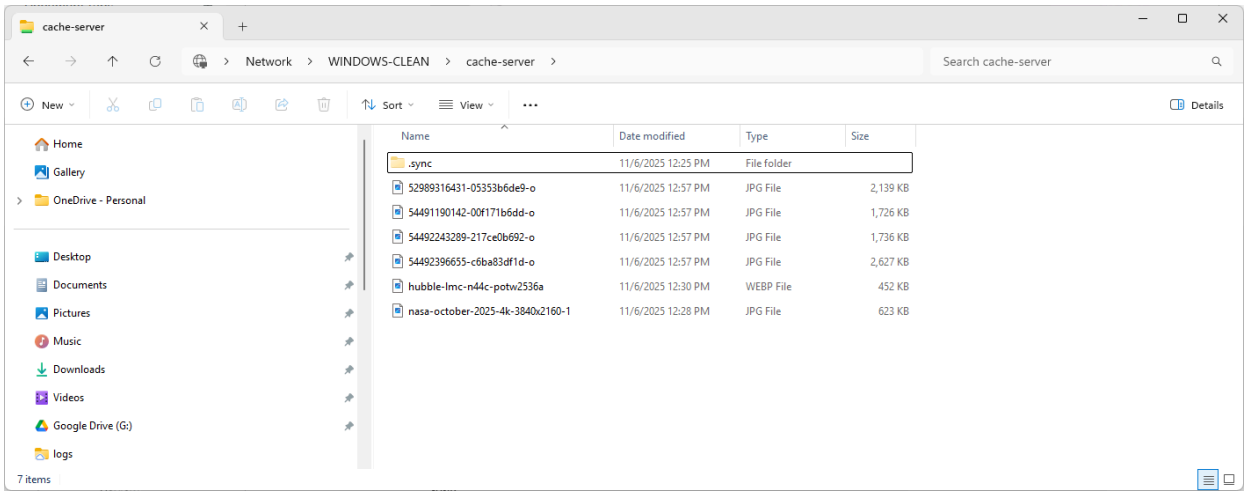

The cloud icon indicates placeholder files and folders.

You can verify that the **Size on disk** is zero bytes by checking the file properties.

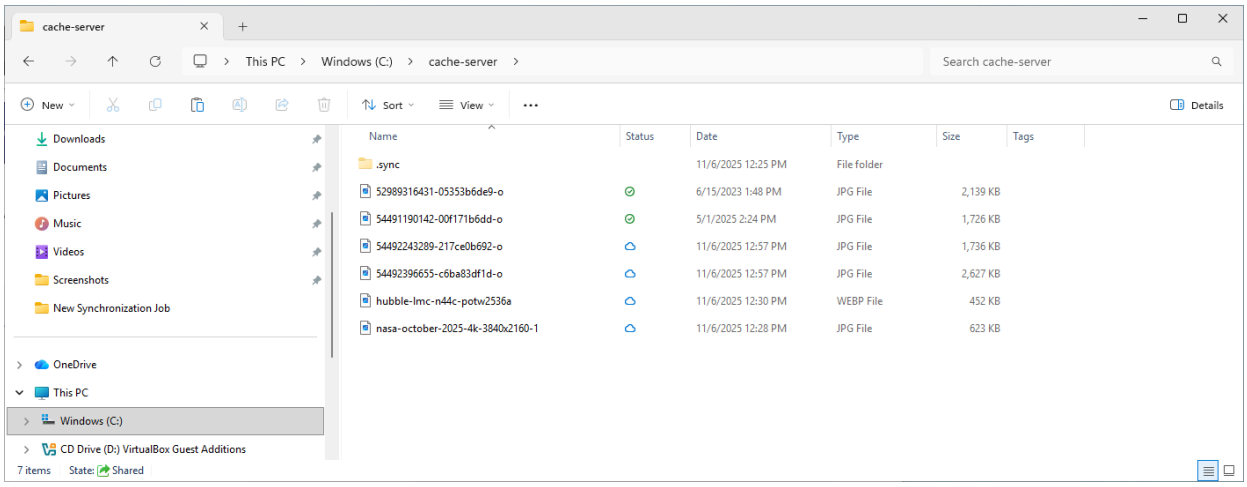



### *The Mounted Folder View*

Another machine that mounts the SMB share should see a list of files and folders like they see on any share.




When files are opened on the mounted folders, they are retrieved by the Caching Gateway from the Primary Storage. In this example, the first two image files (jpg) were opened. The Caching Gateway retrieves the files from the Primary Storage device:

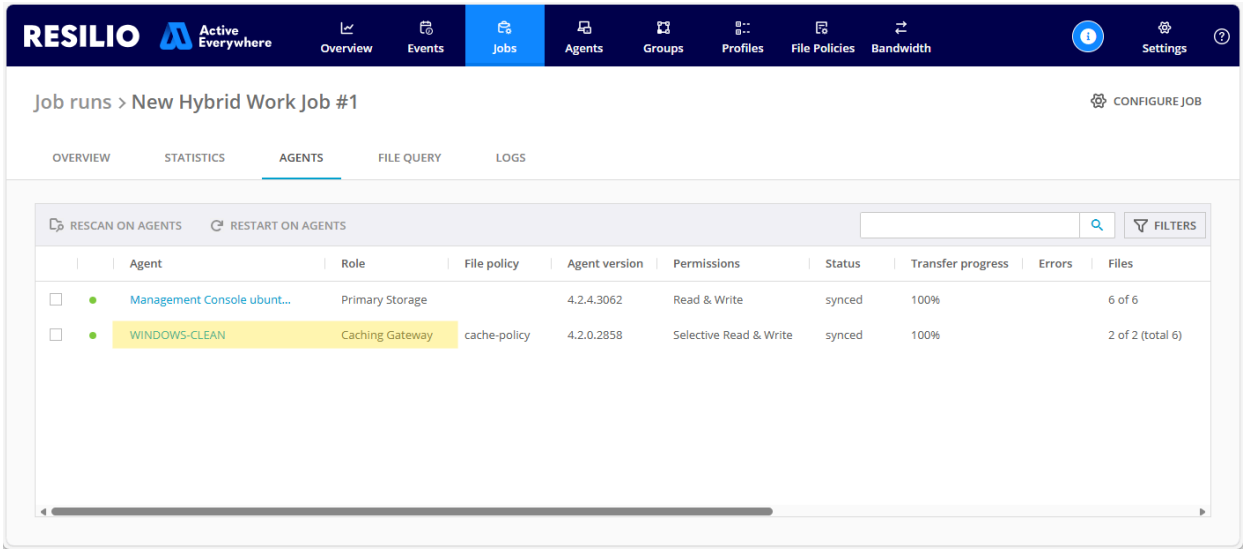



The green checkmark icon indicates that the file is cached on the Caching Gateway.

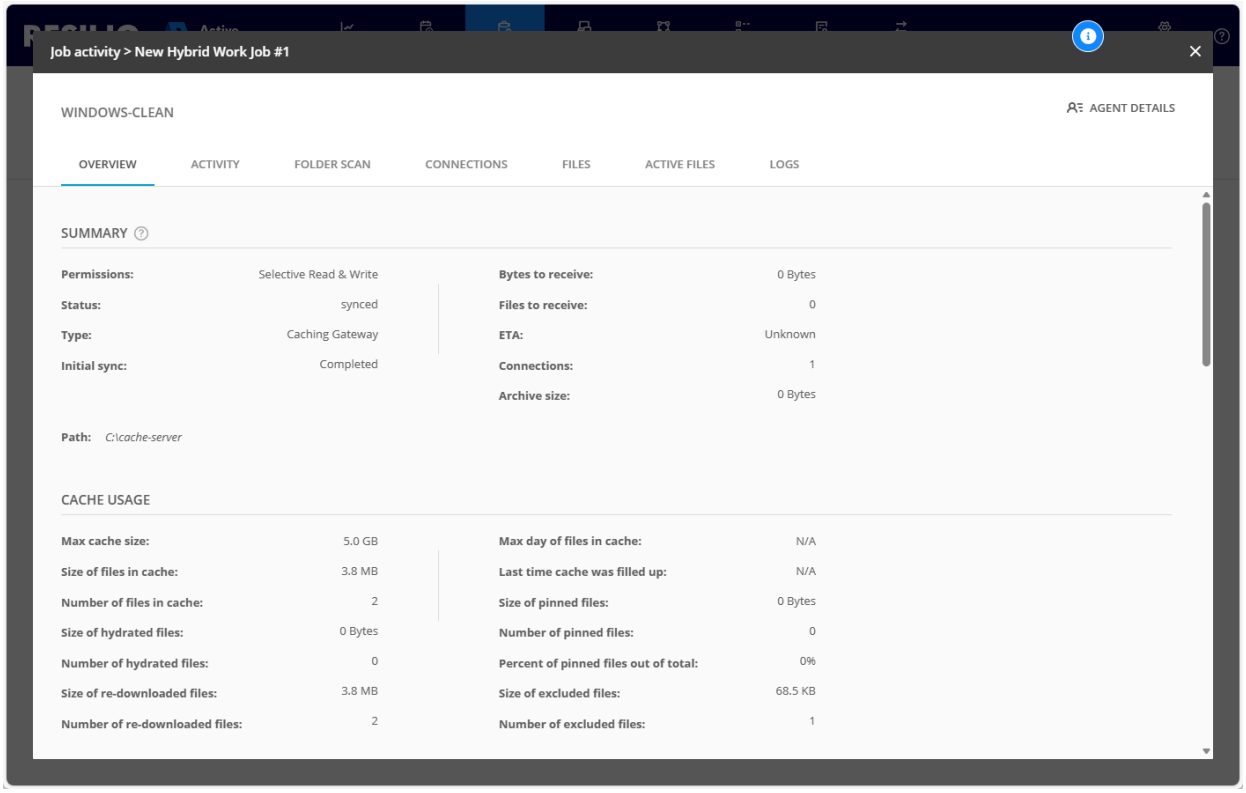
### *The Management Console View*

Resilio's Management Console provides high-level information on the state of each Caching Gateway.

Navigate to **Jobs** > **Job Runs** > **New Hybrid Work Job #1**. Select the **Agents** tab and click the Caching Gateway instance.




It will take you to a view like that where you can see the number of files in the cache, total bytes cached and other statistics.

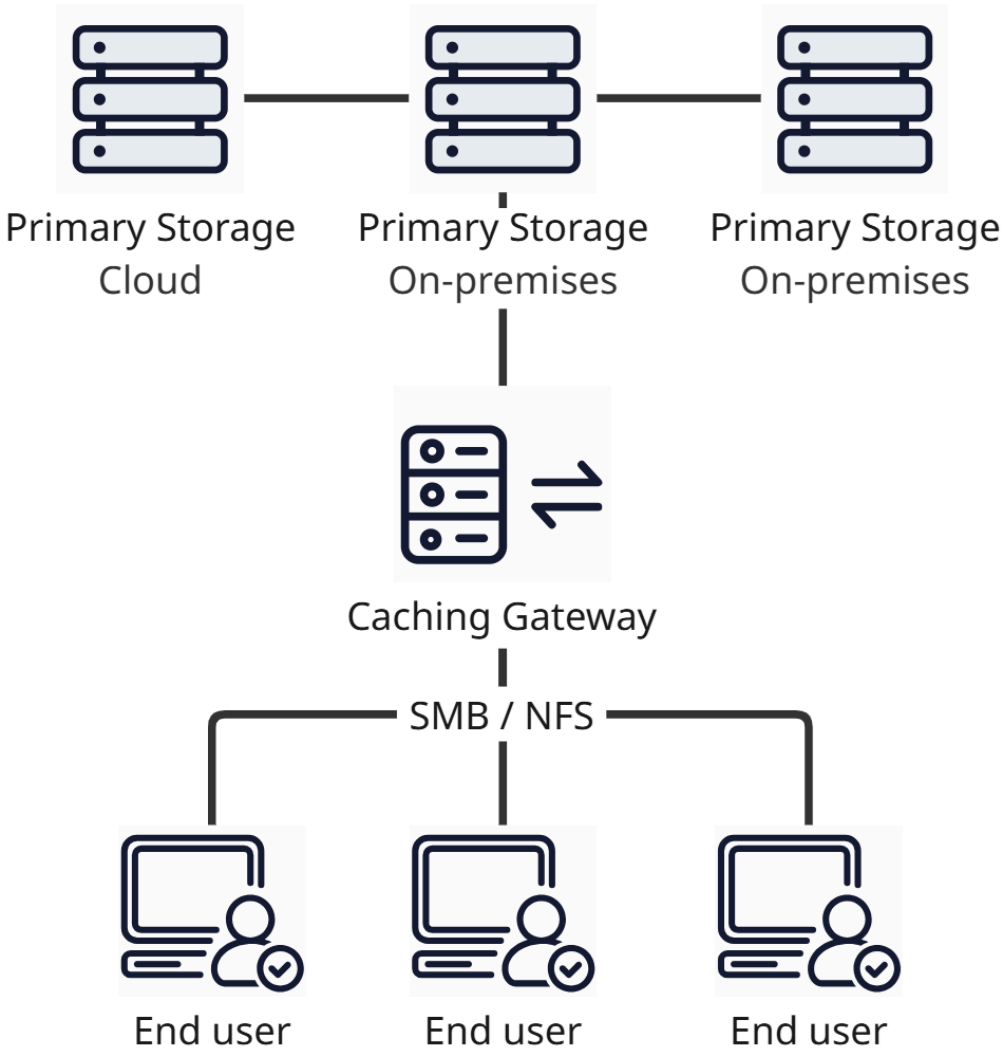


## Adding Data Resiliency and Improving Performance

### Overview

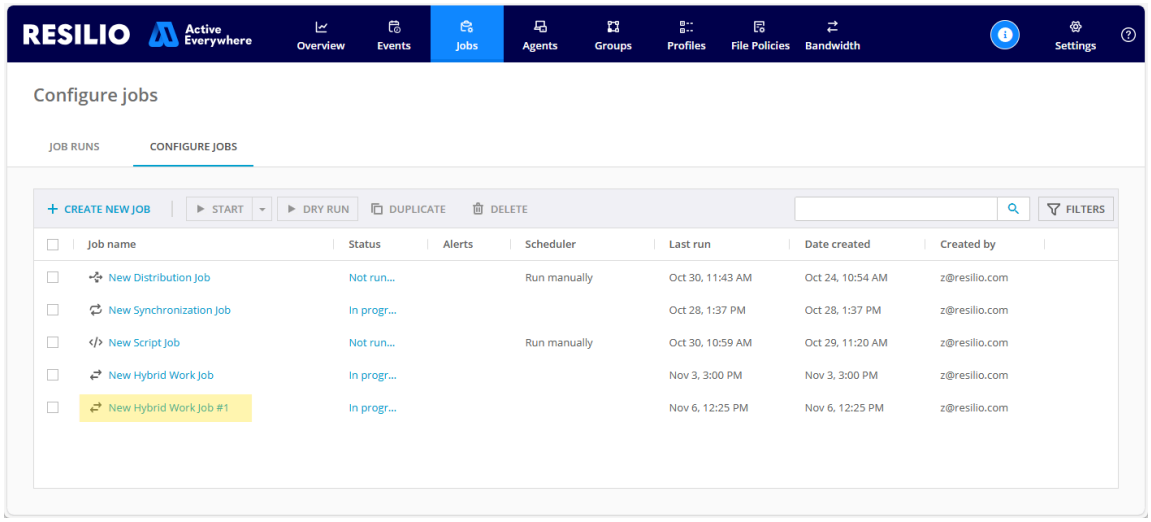
So far, we only configured a single Primary Storage which means we only have a single live copy of our data. This has data availability and performance implications - all Caching Gateways and all end-user devices are accessing a single device for all new data requests.

We will address this issue by adding additional Primary Storage devices. We will add a second server (for example, in a different datacenter) and a third instance that stores all the data in cloud object storage.

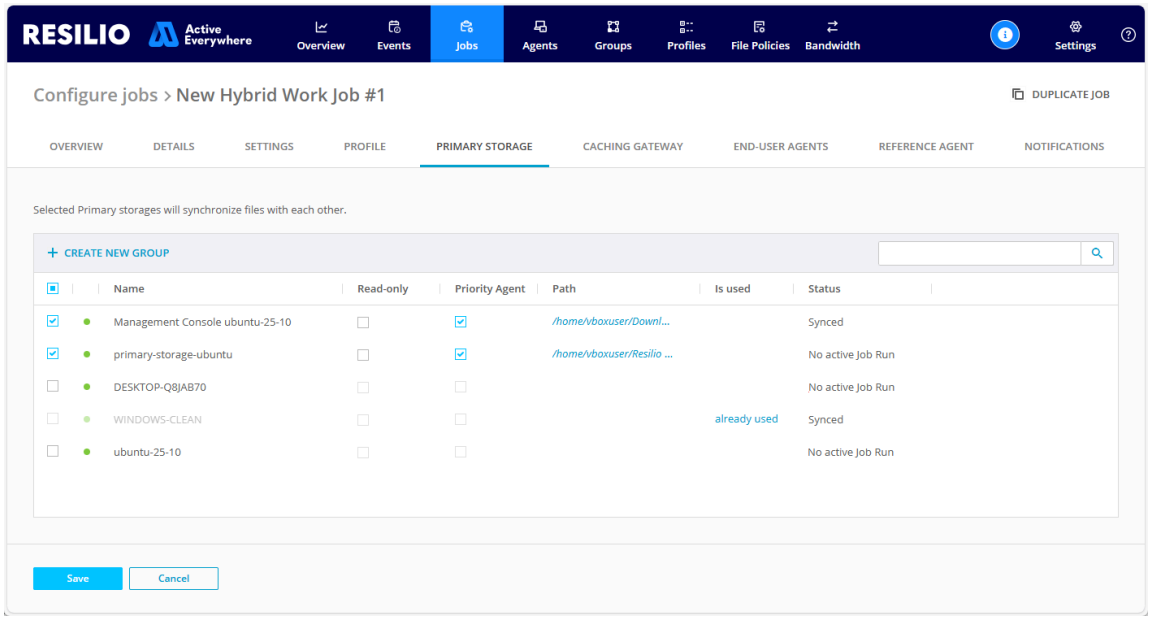


## Configuration

1. Configure a Storage Connector that will be used to facilitate data transfers to and from a cloud storage instance. For details, see [Setting up cloud storage and configuring jobs - Resilio Help Center].
2. Select **Jobs** > **Configure Jobs** and click the previously created Job to edit its configuration.

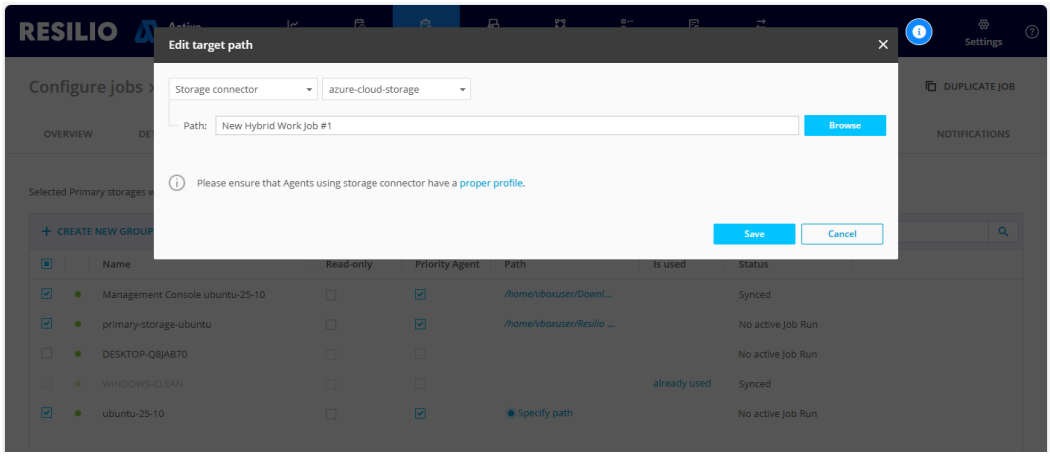



3. In the **Primary Storage** tab, select an Agent instance running the second storage server and set the path to where you want to store a copy of your data.

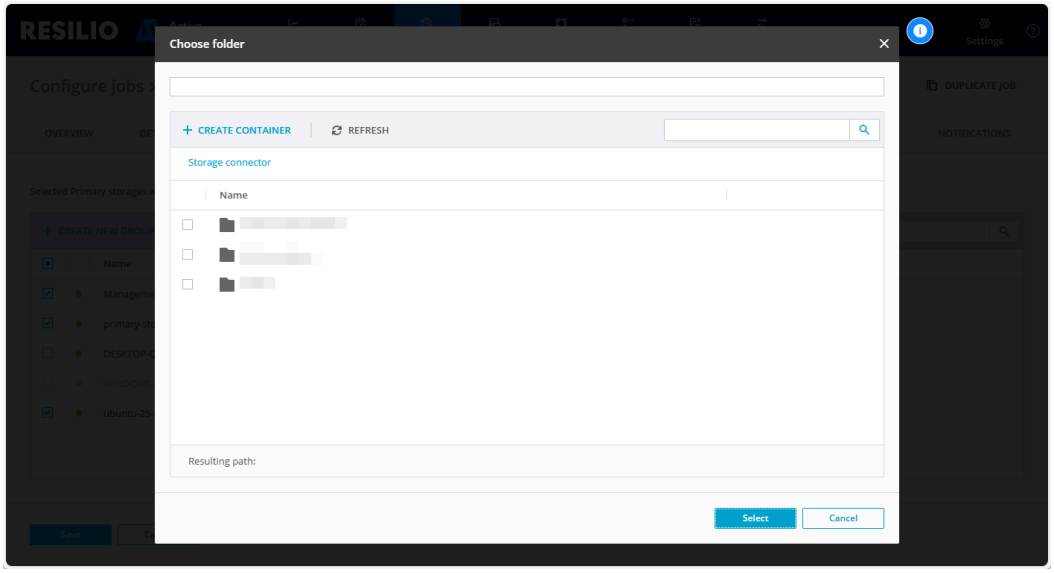



4. Add the third Agent that will act as an interface with the cloud storage:
    a. In the **Primary Storage** tab, select an Agent instance that will be responsible for transferring data to and from a cloud storage.

b. Click the pre-defined value in the path column.
c. From the storage location drop-down list, select **Storage connector**, then choose the previously created cloud storage connector.

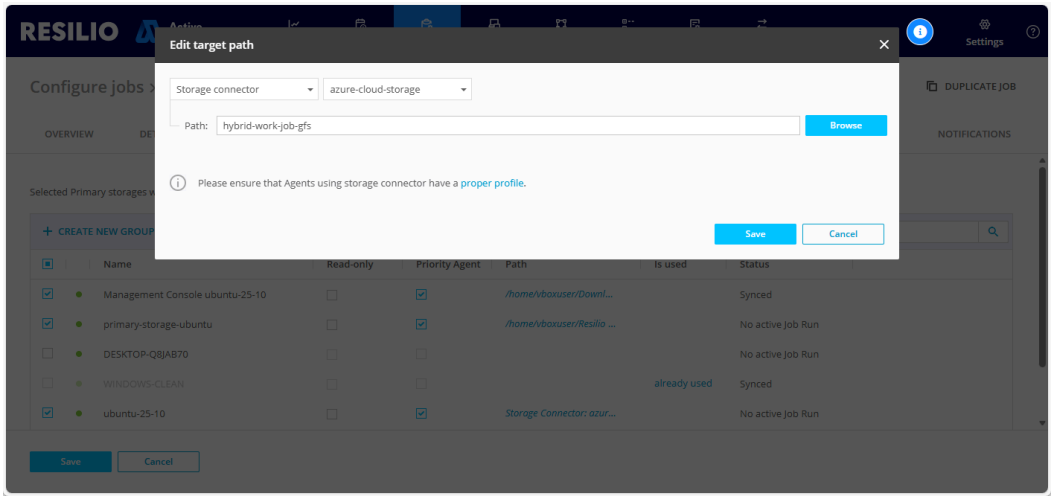



d. Click **Browse**, then click **+ CREATE CONTAINER** or **+ CREATE FOLDER.**

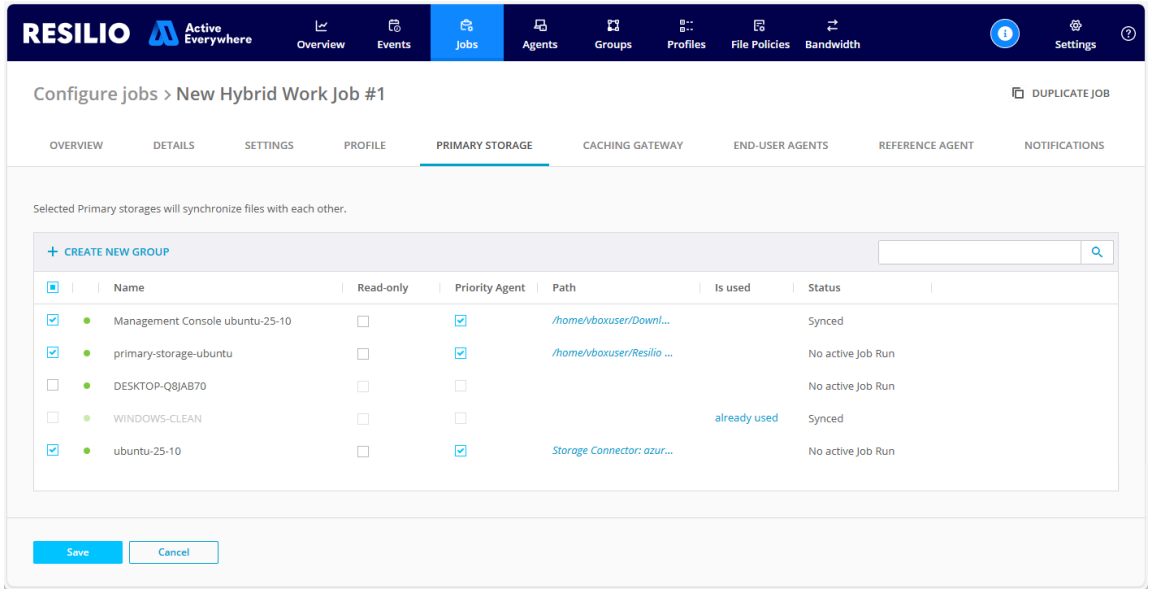



e. Enter the name, then click **Create**.

f. Click **Select** to confirm the storage location, then click **Save** to finish setting up the storage path.

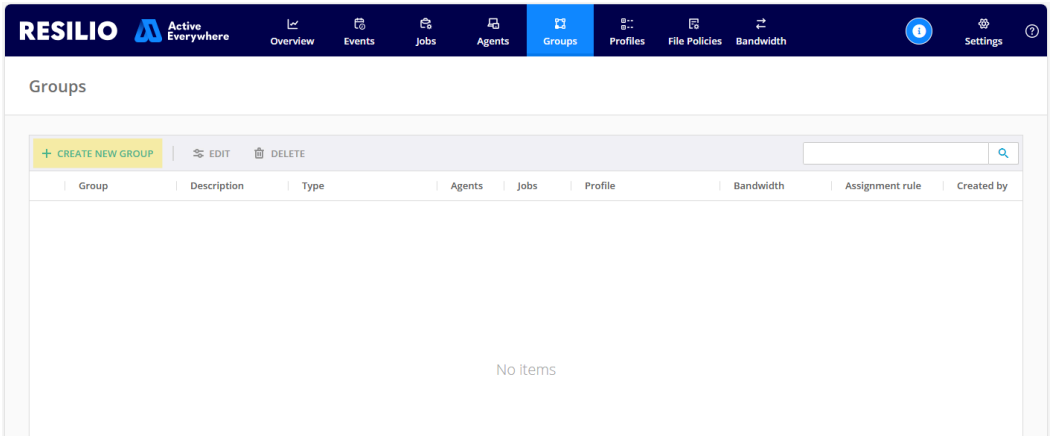



5. Click **Save** in the Job wizard to save your changes.



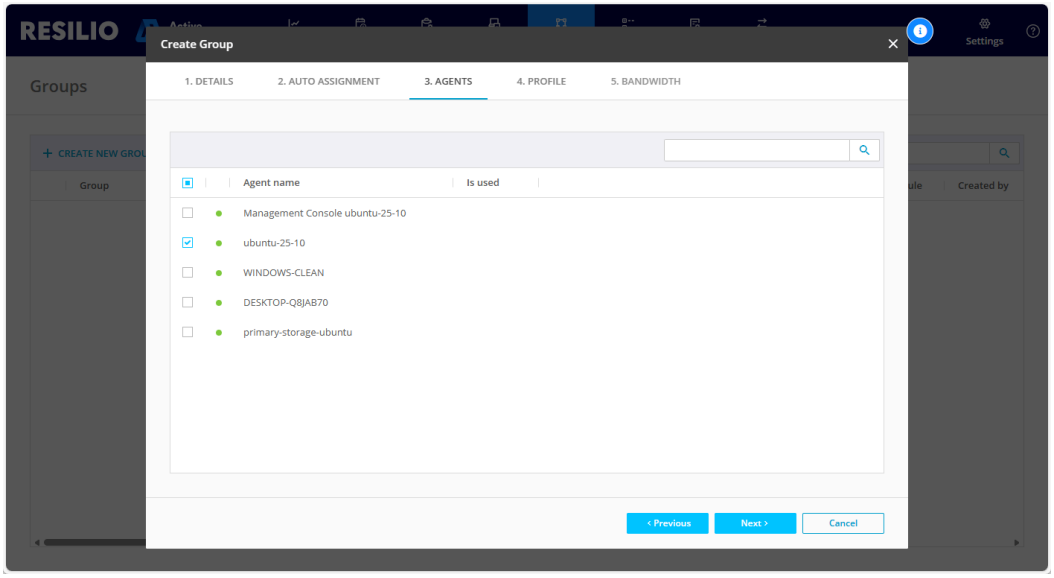

6. To ensure optimal performance of the cloud Agent, assign to it the Storage Connector Agent Profile:

a. Select **Groups**, then click **+ CREATE NEW GROUP**.

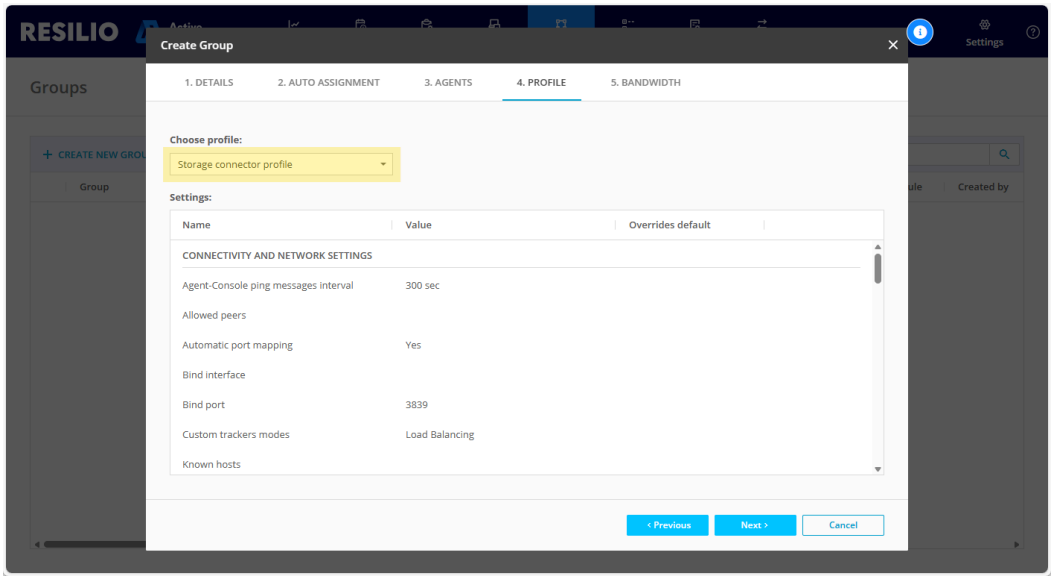



b. Enter the name, then advance to the **Agents** tab.

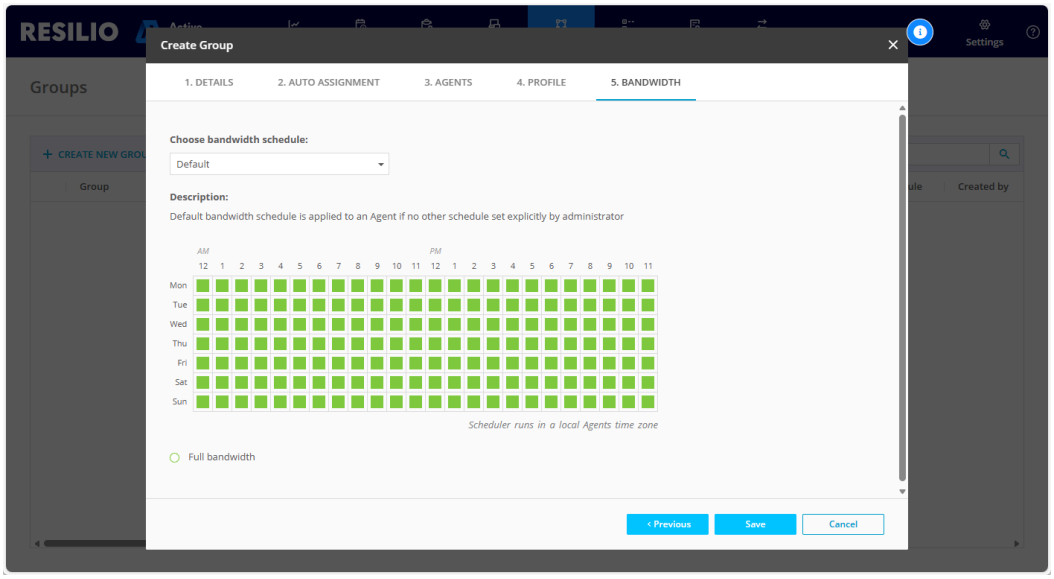
c. Select your cloud Agent instance, then click **Next**.



d. From the **Choose profile** drop-down list, select **Storage connector profile**, then click **Next**.




e. Leave the default bandwidth schedule unchanged to allow unrestricted data transfer speeds, then click **Save**.




## Review

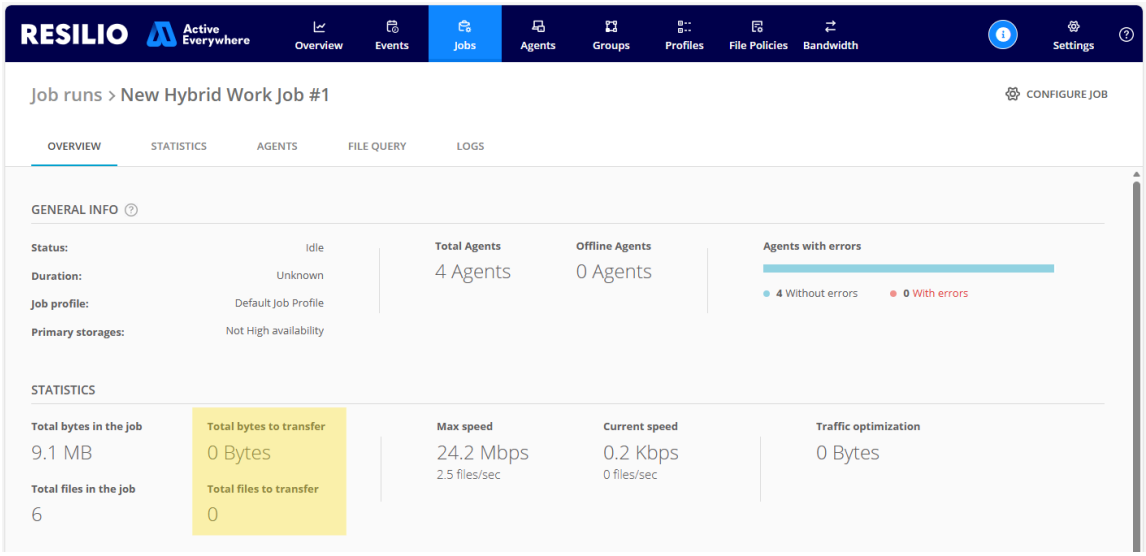
We added data resiliency to our setup and improved performance. Our data is now stored in three different locations: two separate data centers, and a third copy in the cloud. We should also expect better performance: instead of having all file requests handled by a single Primary Storage, we now have three sources that are available to handle incoming requests. They are all active and all data access requests are load-balanced automatically between all instances thanks to the P2P protocol.

## Testing

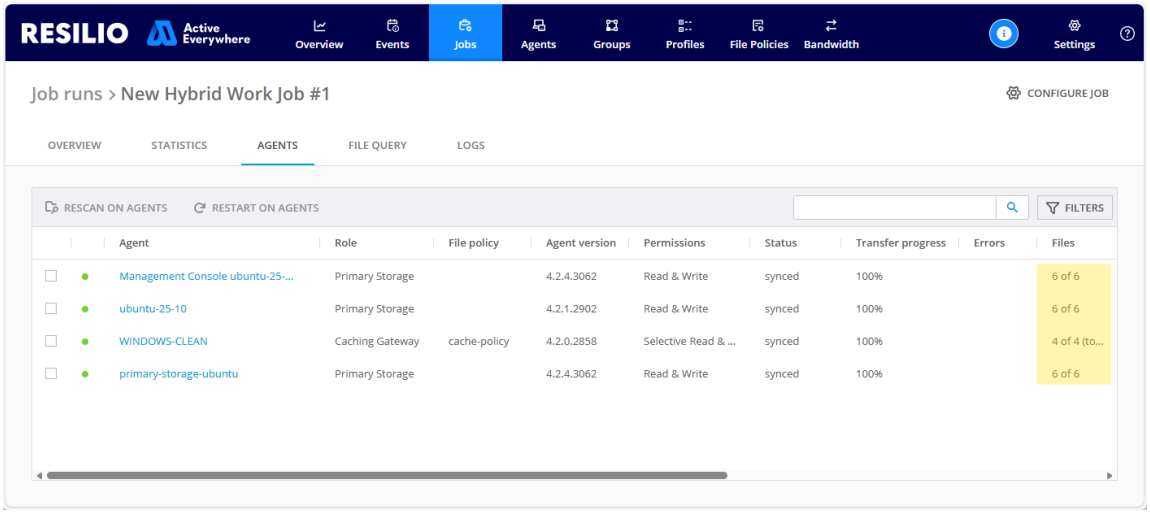
### *The Management Console View*

The Management Console will show you the current state of replication between your three Primary Storages.

1. Select **Jobs** > **Job Runs**, then click the Hybrid Work Job you've created.
2. The **Overview** tab will show you how much data is left to replicate.

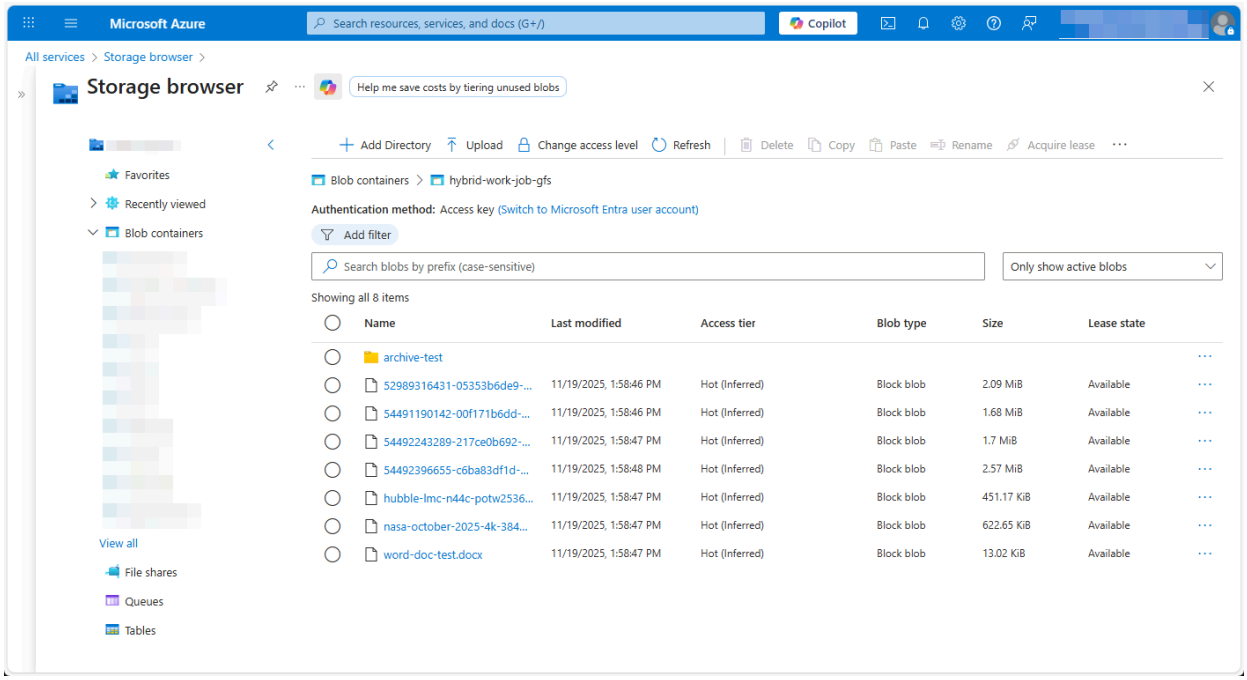



3. Once the replication between the Primary Storages is complete, the **Agents** tab inside the Job Run view will show that all Primary Storages have the same number of files on them.



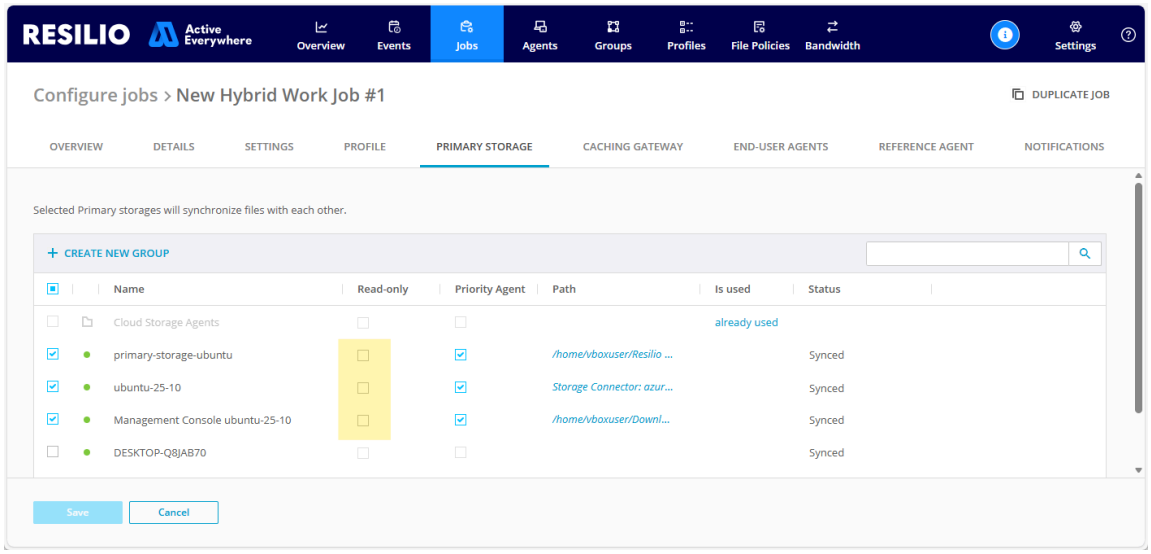

### *The Object Storage View*

Your object storage provider should have a complete copy of your files. Here is an Azure example:




## Further Considerations

1. **Allow read-only access**
   You can choose to make some of your Primary Storages read-only by checking the **Read-only** check box in the **Primary Storage** tab of the Job configuration.




Making a Primary Storage read-only means that no changes from this source will be propagated to other Agents.
For example, if you make your object storage Agent read-only and someone deletes an object using Azure's Storage Browser, this deletion won't be replicated on other Agents.

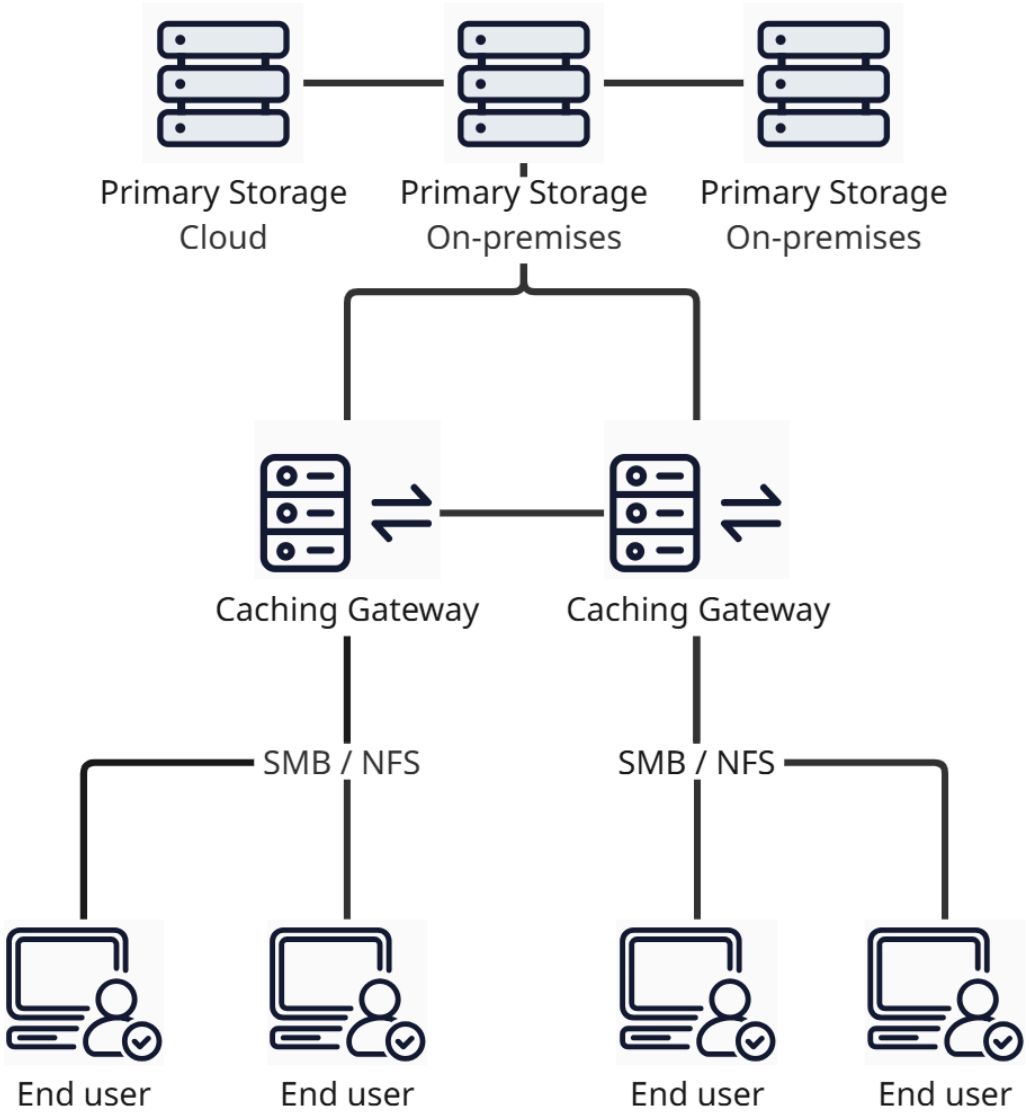
2. **Cloud Agent proximity**
   Performance of the object storage will be better if the Agent is located in a close proximity of the storage instance. For optimal performance, ensure a connection with 10 ms latency or lower.

## Adding Additional Caching Gateways

### Overview

It's time to expand our Caching Gateway infrastructure.  We will spin up additional Caching Gateways in select branch offices. SMB and NFS protocols are highly susceptible to network latency. For best results, deploy Caching Gateways as close to the users as possible.

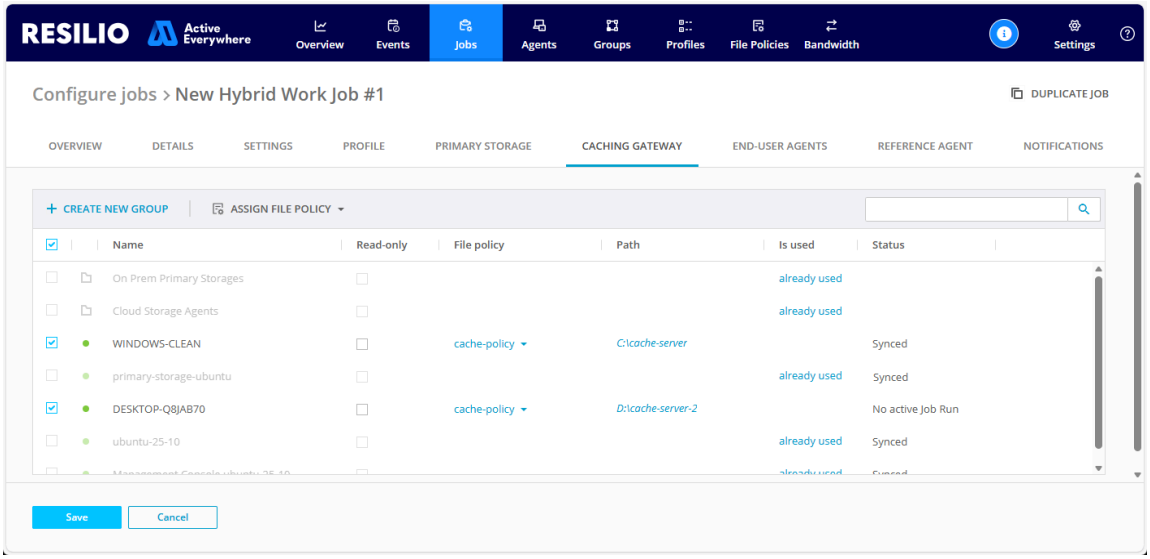
The diagram below shows the updated system architecture.



### Configuration

To add additional Caching Gateways:

1. Select **Jobs** > **Configure Jobs** and click your Hybrid Work job configuration.
2. In the **CACHING GATEWAY** tab, select the additional Caching Gateway, choose a File policy and set up a storage path.




3. Click **Save**.
4. Like before, you'll have to expose the storage folder as an SMB or an NFS share.

### Review

We've added additional Caching Gateways to our setup. They are located in different regions, optimized to serve nearby users and applications. For now, they all use the same caching policy. We will refine our caching policies later.

### Testing

Test your setup similarly to how the single Caching Gateway configuration was tested.

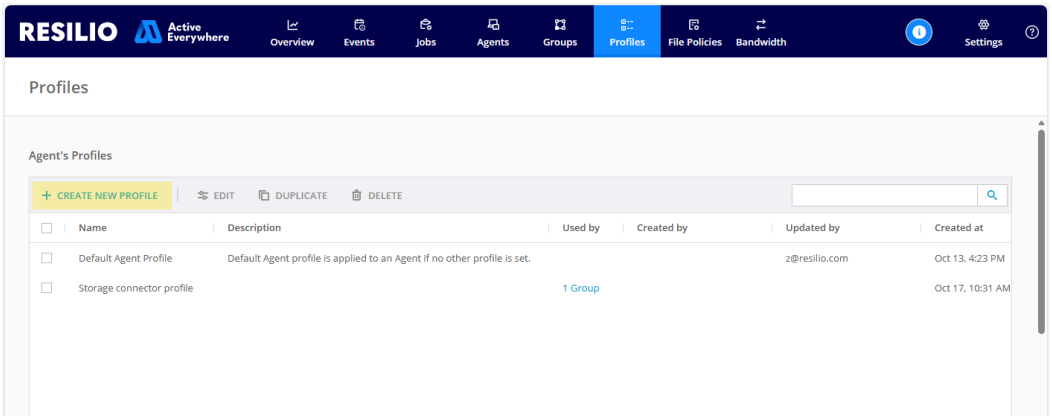
## Revision History

### Overview

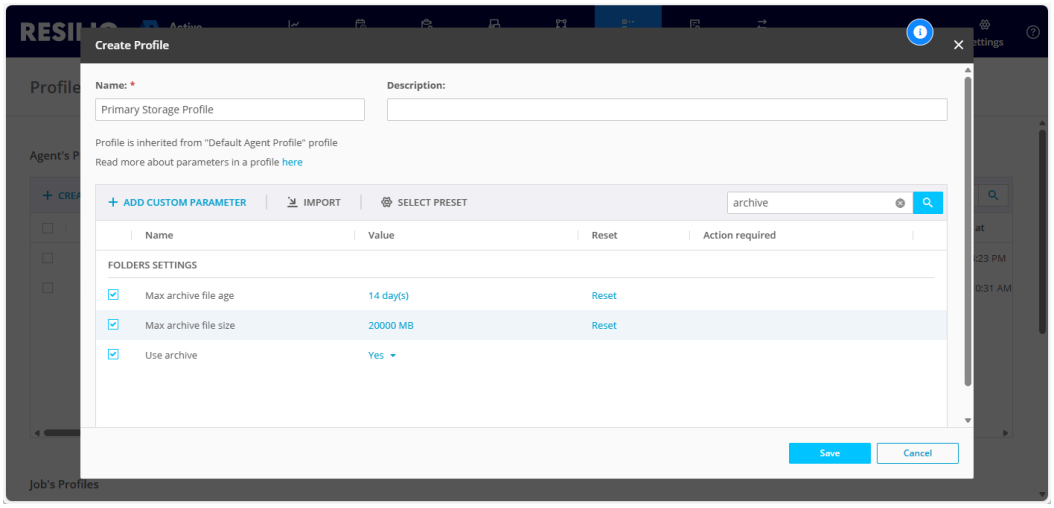
One more aspect of data resiliency is having a readily available file revision history. Resilio has a file archiving feature, and in this section, we will configure and test it. This addition won't require adding new Agents or new storage locations and therefore won't impact our system diagram.

## Configuration

1. Create a dedicated Agent Profile for our on-prem Primary Storages:
    a. Select **Profiles**, then in the **Agent's Profiles** section, click **+ CREATE NEW PROFILE**.



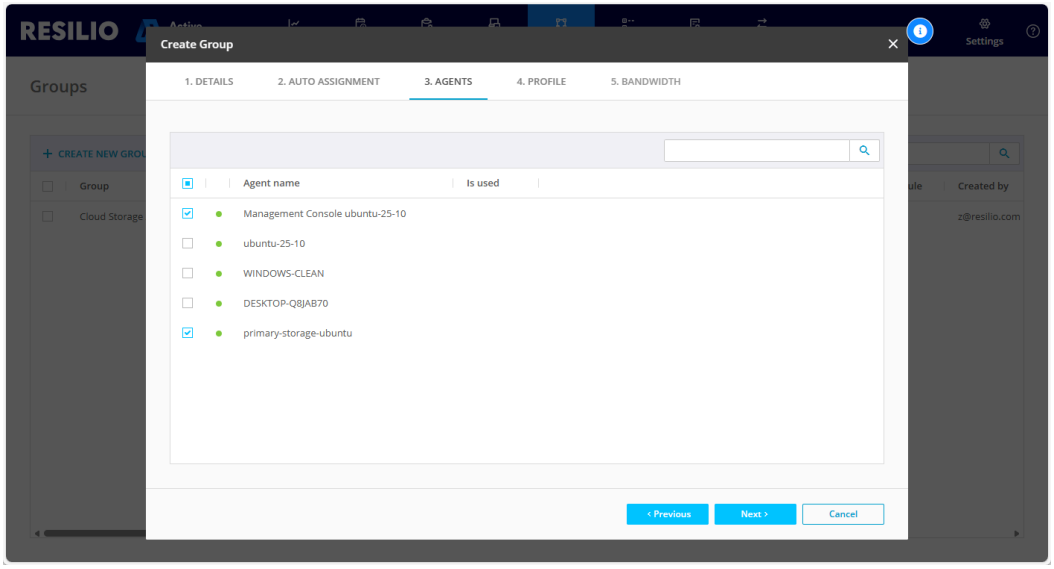

    b. Provide the name for your profile.
    c. In the search field, enter `archive`.
    Set the parameters to 14 days, 20000 MB, and ensure the **Use Archive** option is enabled.
    **Tip**: You can use different values which are more suitable for your use case.



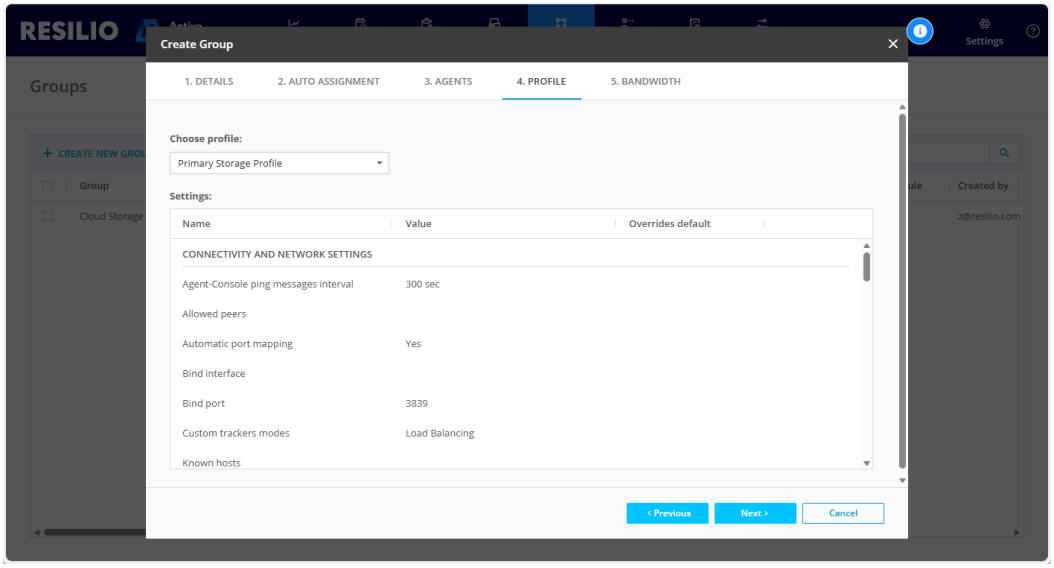

    d. Click **Save**.

2. Associate the newly created Profile with our on-prem Primary Storages:
    a. Select **Groups**, then click **+ CREATE NEW GROUP**.
    b. Enter the name, then advance to the **Agents** tab.

c. Select your two on-prem storage Agents, then click **Next**.




d. From the **Choose profile** drop-down list, select the newly created custom profile, then click **Next**.




e. Leave the default bandwidth schedule unchanged to allow unrestricted data transfer speeds, then click **Save**.

## Review

We configured our two Primary Storage to store a revision history of files for 14 days. The history will include all file versions that were saved and synchronized as well as any file that was deleted. For a thorough description of this feature, see [Understanding the Archive folder - Resilio Help Center].

### Testing

1. Mount the folder of one of your gateways and create a file called “text-1.txt”
2. Open the txt file.
3. Write something to the file and save it.
4. Write some more and save it again.
5. Go to one of your Primary Storage servers and browse the file system to <path you selected for your Primary Storage>/.sync/Archive
6. The folder should contain multiple files with suffixes for each file revision:

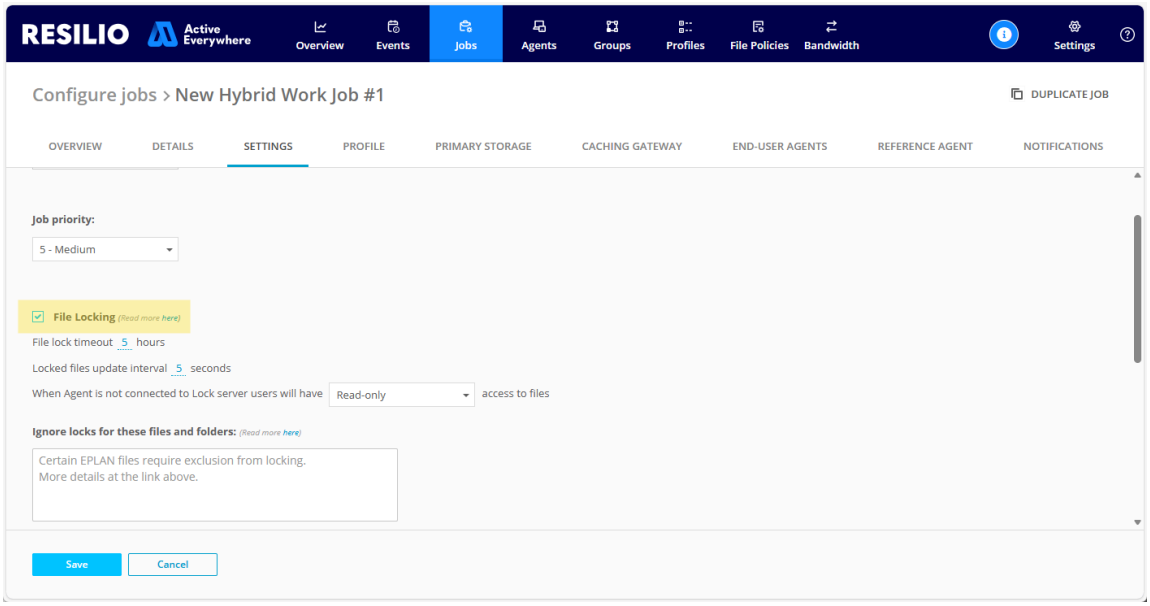

## Distributed File Locking

### Overview

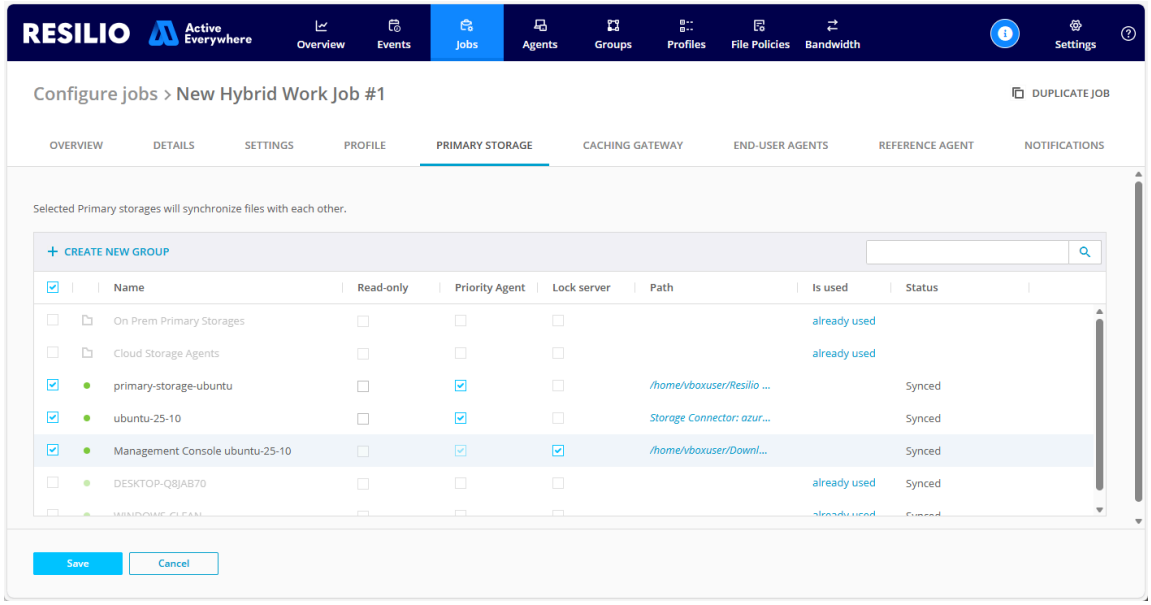
Having multiple Caching Gateways means that there are multiple access paths to each file and the central file server is no longer aware of concurrent editing attempts. This allows multiple users or applications to access and edit the same file at the same time, which may result in multiple users overwriting each other's updates. To mitigate that, you can enable file locking.

### Configuration

1. Select **Jobs** > **Configure Jobs** and click your Hybrid Work job definition.
2. In the **SETTINGS** tab, select the **File locking** option.



3. In the **PRIMARY STORAGE** tab, choose one of your Primary Storage devices to be the **Lock server**.




4. Click **Save**.

## Review

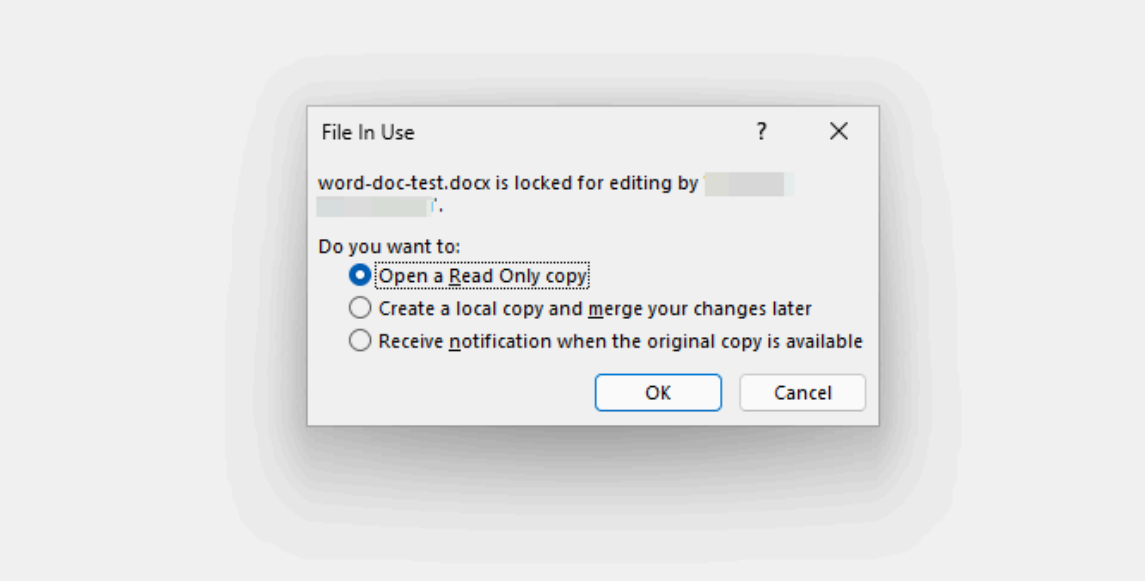
With this change, only one editor will be allowed at any given time. Note that the locking behavior depends on the application - the application needs to ask for a lock for Resilio to enforce the distributed locking mechanism. Simple applications, like Notepad on Windows, don't acquire a lock. The overall behavior is similar to how these applications behave when using a central file server.

## Testing

### *The end-user view*

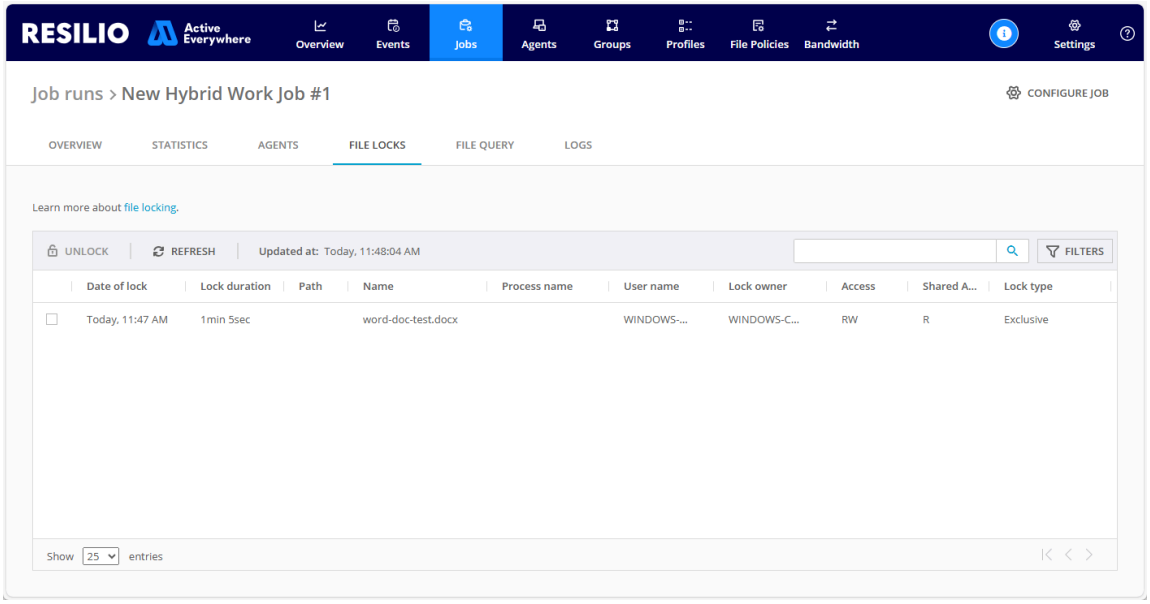
1. Go to a device that mounts one of your Caching Gateways and create a Word document in the shared folder.
2. Open, add some content, then save the file, but don't close it.
3. Go to a different device that mounts another Caching Gateway (not the same one from step 1), and try to open the same file.

4. Microsoft Word will display a **File in Use** prompt.




*The Management Console view*

1. In the Management Console, select **Jobs** > **Job Runs**, then click your Hybrid Work Job definition.
2. Select the FILE LOCKS tab to review current file locks.



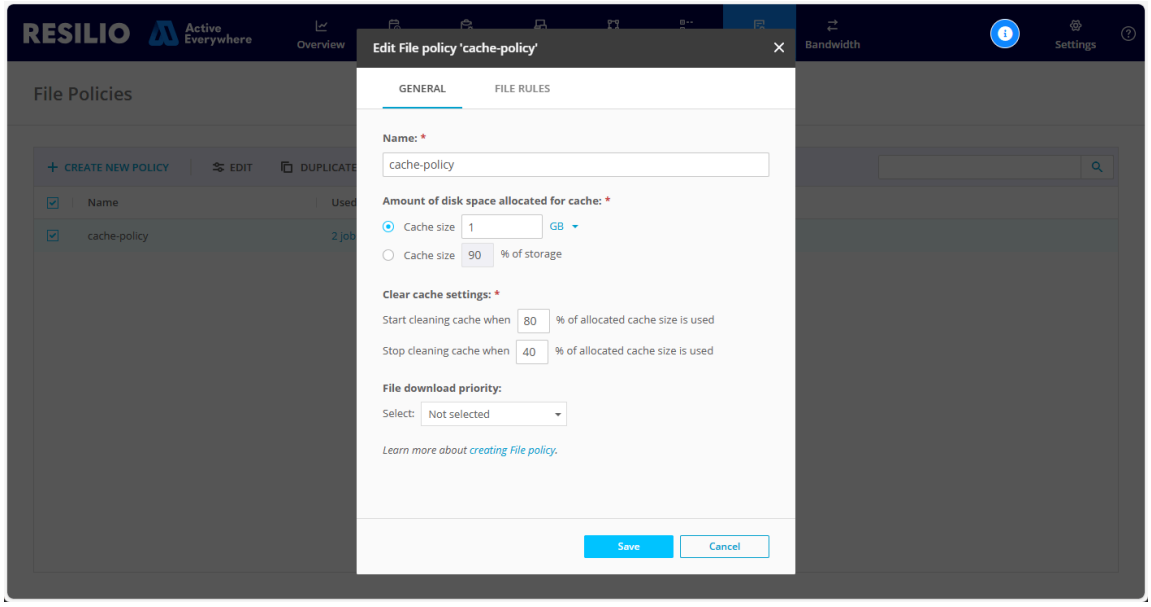
## Cache Sizing

### Overview

In this section we will look into cache sizing, the cache behavior when it is getting full and learn more about the caching stats.

### Configuration

1. To make it easier to simulate a full cache, let's make our cache very small.
2. Go to File Policies and edit the cache policy we used for the Caching Gateways - `cache-policy`.
3. Set it to the following values:

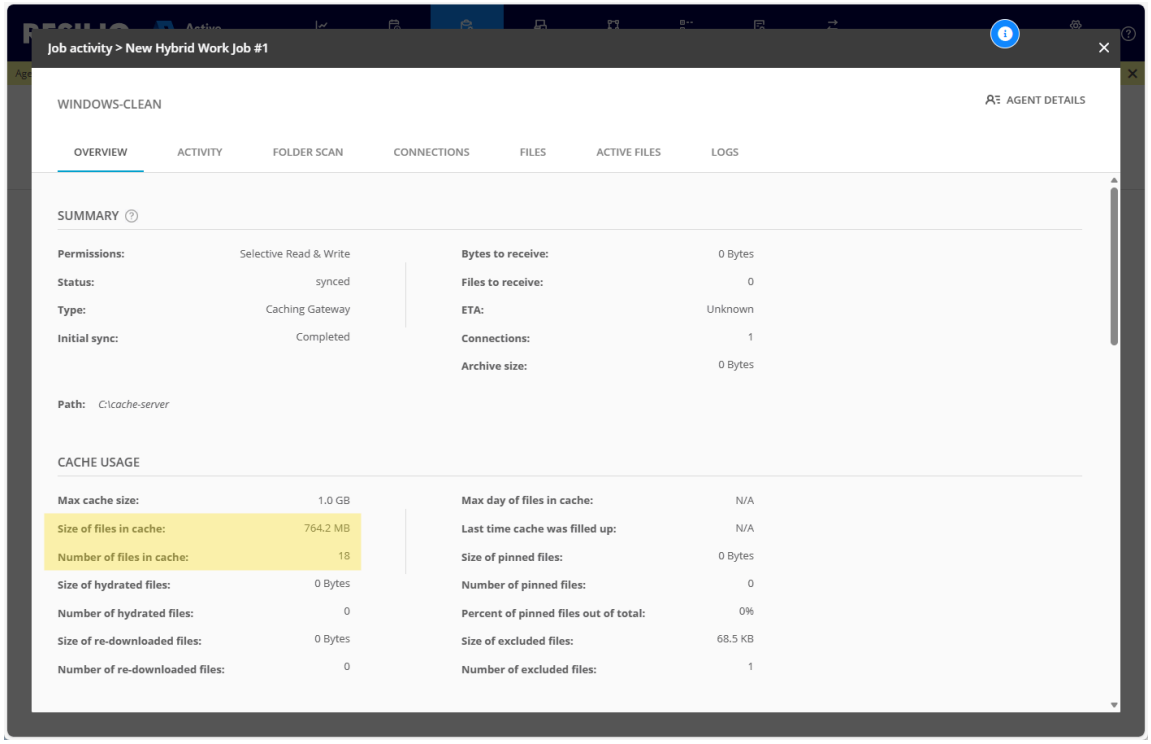



4. Save it.

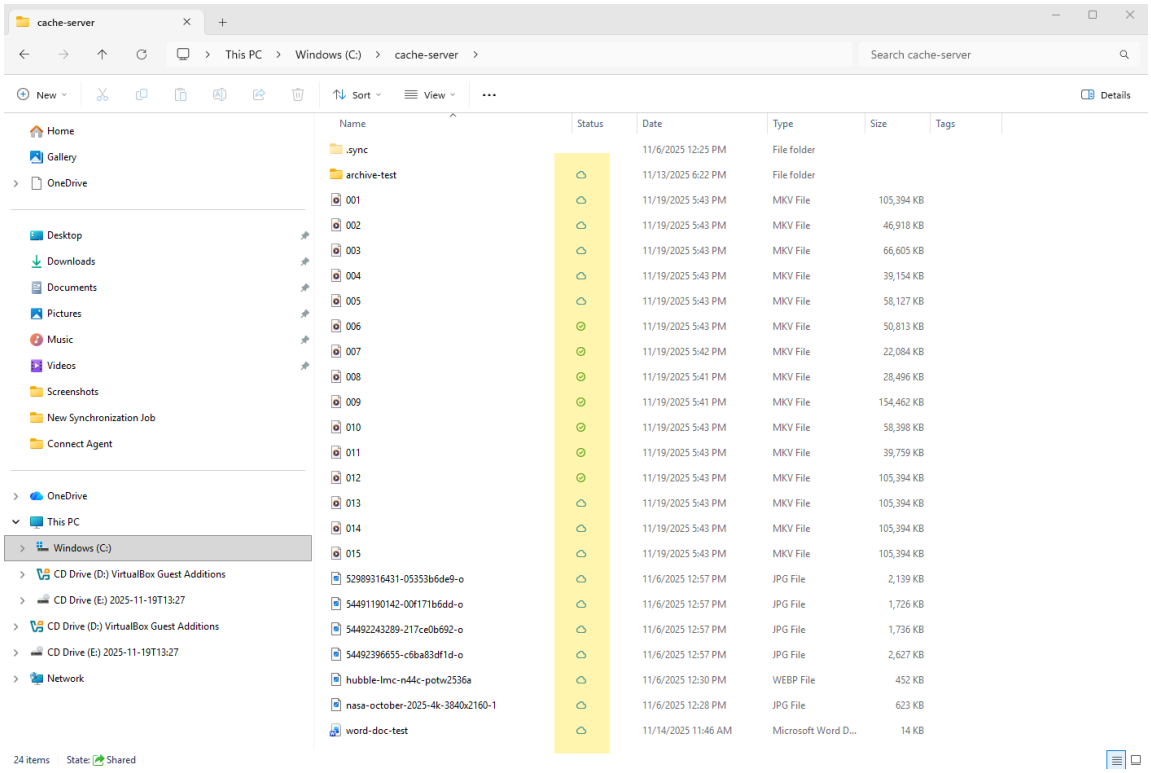
### Testing

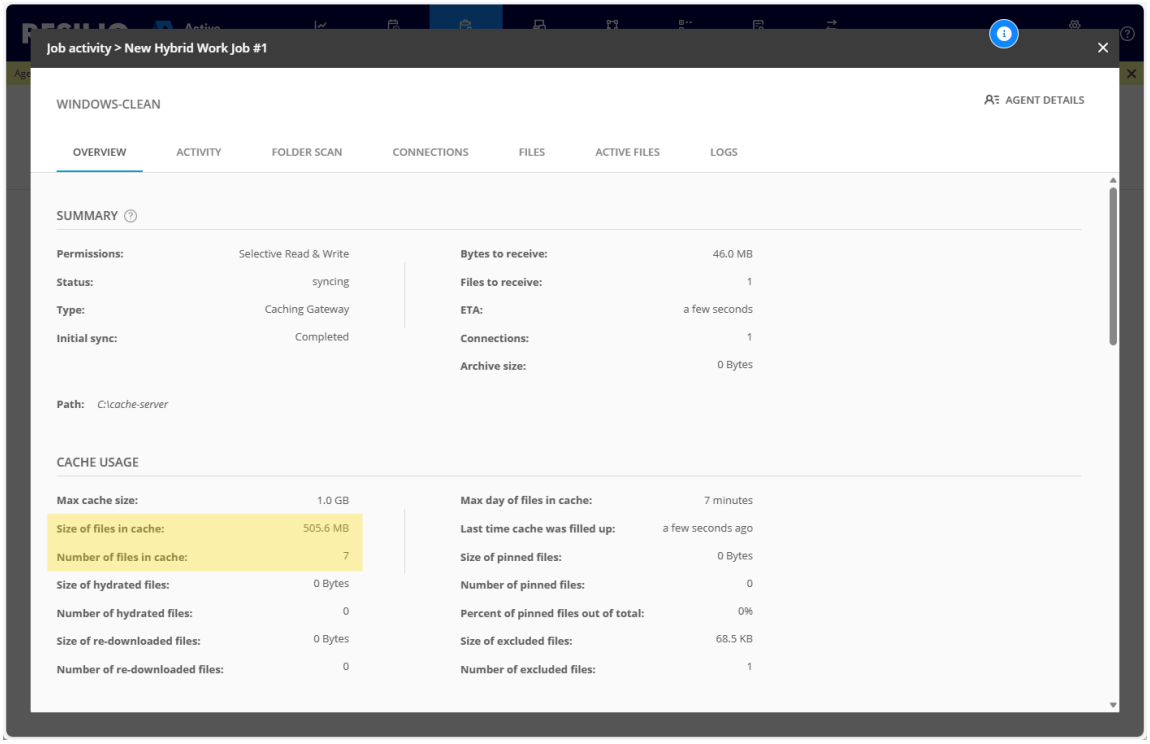
5. 
6. Populate the Primary Storage with 1 GB of data, preferably with larger files.

7. Start hydrating the cache by opening files on the Caching Gateway over SMB.




8. Notice how after exceeding 800 MB, the Agent on the Caching Gateway will start freeing up the storage space occupied by cached files. The dehydration queue is based on access time - files with the oldest timestamp are dehydrated first.


cache-server
This PC
Windows (C:)
cache-server
Search cache-server
New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
New Synchronization Job
Connect Agent
OneDrive
This PC
Windows (C:)
CD Drive (D:) VirtualBox Guest Additions
CD Drive (E:) 2025-11-19T13:27
CD Drive (D:) VirtualBox Guest Additions
CD Drive (E:) 2025-11-19T13:27
Network
Name
Status
Date
Type
Size
Tags
.sync 11/6/2025 12:25 PM File folder
archive-test 11/13/2025 6:22 PM File folder
001 11/19/2025 5:43 PM MKV File 105,394 KB
002 11/19/2025 5:43 PM MKV File 46,918 KB
003 11/19/2025 5:43 PM MKV File 66,605 KB
004 11/19/2025 5:43 PM MKV File 39,154 KB
005 11/19/2025 5:43 PM MKV File 58,127 KB
006 11/19/2025 5:43 PM MKV File 50,813 KB
007 11/19/2025 5:42 PM MKV File 22,084 KB
008 11/19/2025 5:41 PM MKV File 28,496 KB
009 11/19/2025 5:41 PM MKV File 154,462 KB
010 11/19/2025 5:43 PM MKV File 58,398 KB
011 11/19/2025 5:43 PM MKV File 39,759 KB
012 11/19/2025 5:43 PM MKV File 105,394 KB
013 11/19/2025 5:43 PM MKV File 105,394 KB
014 11/19/2025 5:43 PM MKV File 105,394 KB
015 11/19/2025 5:43 PM MKV File 105,394 KB
52989316431-05353b6de9-o 11/6/2025 12:57 PM JPG File 2,139 KB
54491190142-00f171b6dd-o 11/6/2025 12:57 PM JPG File 1,726 KB
54492243289-217ce0b692-o 11/6/2025 12:57 PM JPG File 1,736 KB
54492396655-c6ba83df1d-o 11/6/2025 12:57 PM JPG File 2,627 KB
hubble-lmc-n44c-potw2536a 11/6/2025 12:30 PM WEBP File 452 KB
nasa-october-2025-4k-3840x2160-1 11/6/2025 12:28 PM JPG File 623 KB
word-doc-test 11/14/2025 11:46 AM Microsoft Word D... 14 KB
24 items
State: Shared


Job activity > New Hybrid Work Job #1
WINDOWS-CLEAN
AGENT DETAILS
OVERVIEW
ACTIVITY
FOLDER SCAN
CONNECTIONS
FILES
ACTIVE FILES
LOGS
SUMMARY
Permissions: Selective Read & Write
Status: syncing
Type: Caching Gateway
Initial sync: Completed
Bytes to receive: 46.0 MB
Files to receive: 1
ETA: a few seconds
Connections: 1
Archive size: 0 Bytes
Path: C:\cache-server
CACHE USAGE
Max cache size: 1.0 GB
Size of files in cache: 505.6 MB
Number of files in cache: 7
Size of hydrated files: 0 Bytes
Number of hydrated files: 0
Size of re-downloaded files: 0 Bytes
Number of re-downloaded files: 0
Max day of files in cache: 7 minutes
Last time cache was filled up: a few seconds ago
Size of pinned files: 0 Bytes
Number of pinned files: 0
Percent of pinned files out of total: 0%
Size of excluded files: 68.5 KB
Number of excluded files: 1

Review

With File Policies, you can optimize the use of available storage space while ensuring that frequently accessed files remain available on the Caching Gateway through access-time-based dehydration.
You can create multiple policies tailored to the needs of specific office locations and their hardware capabilities.

## File Pinning

### Overview

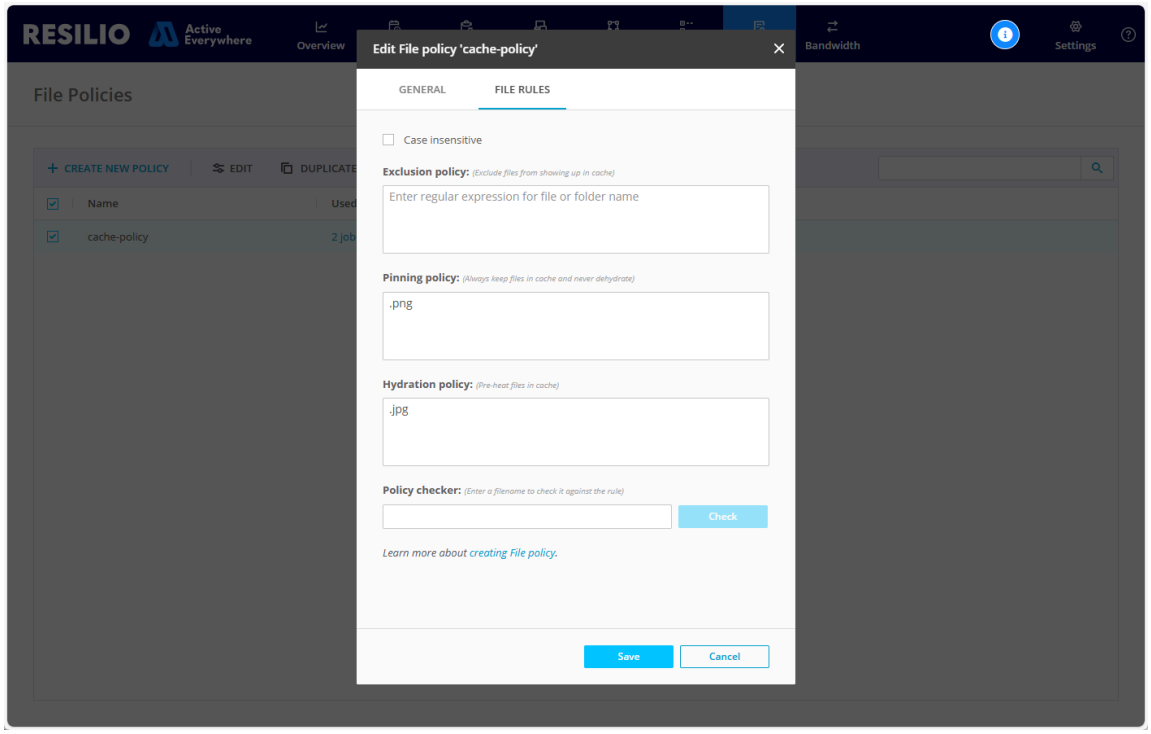
In this section, we will explore how the hydration policy automatically downloads files to Caching Gateways, ensuring they are instantly available for end users. We will also cover how to configure a pinning policy to prevent specific files from being dehydrated.

### Configuration

Let’s edit the previously created File Policy and add a pinning policy and hydration policy.

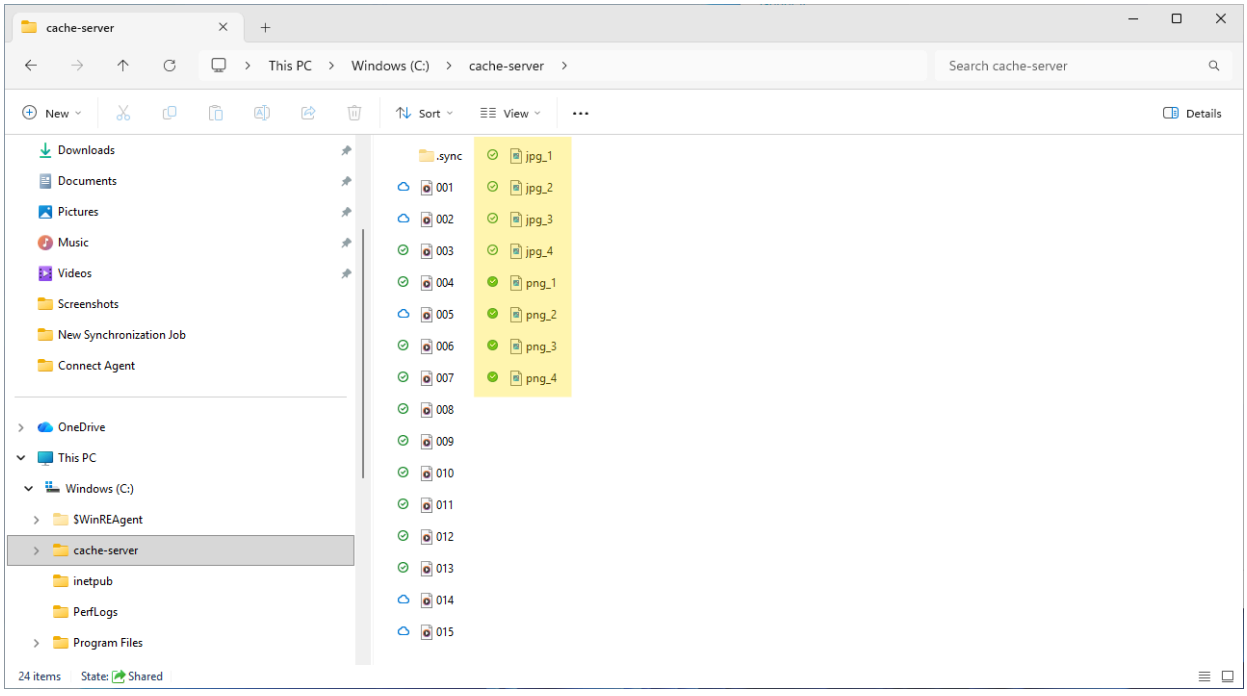
1. Go to File Policies and edit the cache policy we used for the Caching Gateways - cache-policy.
2. In the File Rules tab, add a pinning policy rule that will make sure PNG files always kept in the cache:
   ```
   .png
   ```
3. Add a hydration policy rule that will automatically download JPG files from the Primary Storage device:
   ```
   .jpg
   ```

4. Click Save.




## Testing

Add a few PNG and JPG image files. Notice how JPG as well as PNG files download automatically to your Caching Gateways, with each file type having a different status icon.



Pinned PNG files have a green background checkmark icon, meaning that they will be always available on the Caching Gateway and won't undergo dehydration.
JPG files have a clear background checkmark icon, meaning that they have been downloaded from the Primary Storage (pre-heated), but they will be dehydrated according to the cache clearing settings.

#### Review

By adding pinning and hydration policies you can optimize your cache usage and improve your end user experience.

### Synchronizing ACLs

#### Overview

In this section, we will look into how Resilio Active Everywhere not only synchronizes the files themselves but also file access permissions.

**Prerequisite**:
ACLs synchronization requires users accounts to be managed via Active Directory.

#### Configuration

With the file permissions synchronization enabled in the Job Profile, no additional configuration is necessary.

#### Testing

On a Caching Gateway, pick a file and change its permissions. Observe how the change propagates to other Job participants.

#### Review

ACLs synchronization ensures that file access on all participating devices is restricted to authorized users.

## Conclusion

Resilio Active Everywhere presents a technically superior GFS solution for organizations seeking high performance, scalability, and control over their data. Its server caching capabilities and P2P architecture offer distinct advantages over traditional cloud-centric solutions. For IT professionals and DevOps teams, Resilio provides the tools to build a resilient, efficient, and secure file system tailored to the demands of modern enterprise environments.

## Getting Started

Interested in implementing Resilio Active Everywhere as your GFS solution?

1. Request a Demo: See Resilio Active Everywhere in action tailored to your use case.
2. Pilot Deployment: Start with a proof of concept in your environment.
3. Consultation: Engage with Resilio's technical team to design a solution that fits your needs.

Feel free to explore more about Resilio's capabilities and how it can transform your organization's file synchronization strategy. Whether you're looking to overcome the limitations of cloud-dependent systems or seeking a scalable solution that grows with your enterprise, Resilio Active Everywhere offers a path forward.