Aparavi Active Archive

Go ahead, keep your data.

Just do it better.

February 2018

APARAVI™

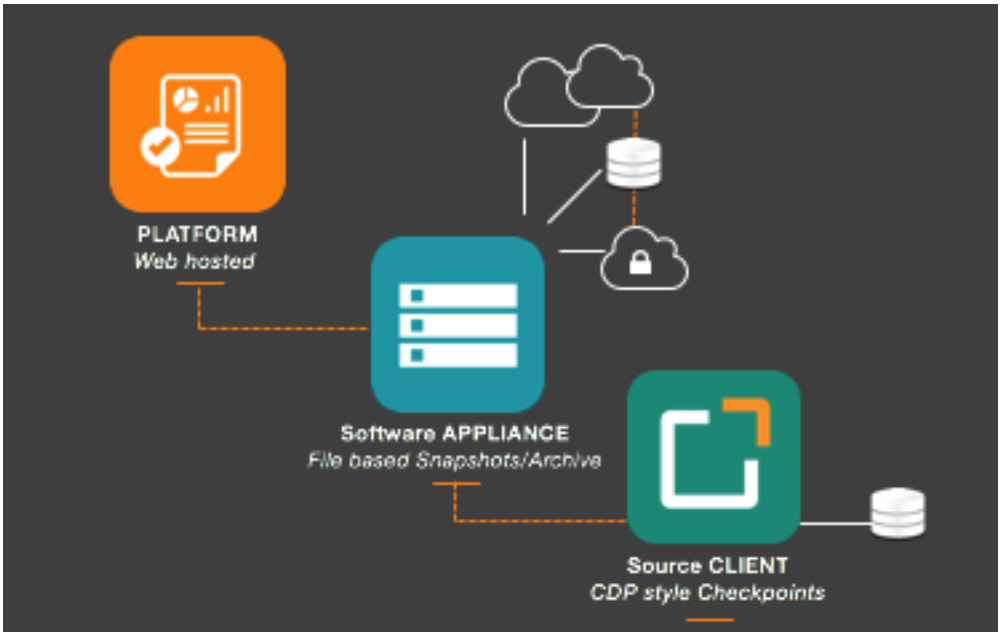
Table f Contents

Aparavi provides a new and much better way to look at data retention. Our SaaS-based active archive solution delivers true storage independence with on-premises and multi-cloud mobility. Drastically reducing the amount of storage required with patent-pending data pruning technology, we also ensure corporate governance. Our technology keeps data only for the period you need it and removes data after it has expired. Working across local and cloud storage destinations and with an open data format, we prevent vendor lock-in for long-term flexibility.

This paper explores how Aparavi active archive works and what typical data management scenarios look like.

I. 3-Tier Architecture and Storage Model

Aparavi active archive consists of a three-tiered architecture and storage model. The 3-tier architecture consists of agents, a software appliance, and a Web app. These work in concert with the storage model which consists of checkpoints, snapshots, and archives.



3-Tier Architecture

Agents

Agents are the data sources. Agents are typically file servers that store a wide variety of data such as pdfs, spreadsheets, images, and much more. These servers contain data that needs to be archived and retained. For maximum data security, all data is encrypted and compressed before it leaves the agent and passed to the software appliance. This is referred to as source node encryption. Agents perform checkpoints which we will discuss in detail.

Software Appliance

Software appliances are usually a local, on premises machine that stores data coming from the agents. An appliance can store any number of copies of agent data, but usually only a few copies are kept, and these are referred to as snapshots. These copies can be utilized to quickly recover data in case of agent failure or loss of recent data.

Software appliances are also responsible for sending data to a long-term archive device, such as a centralized data center (in case of a private cloud) or to a public cloud like Amazon S3 or Microsoft Azure.

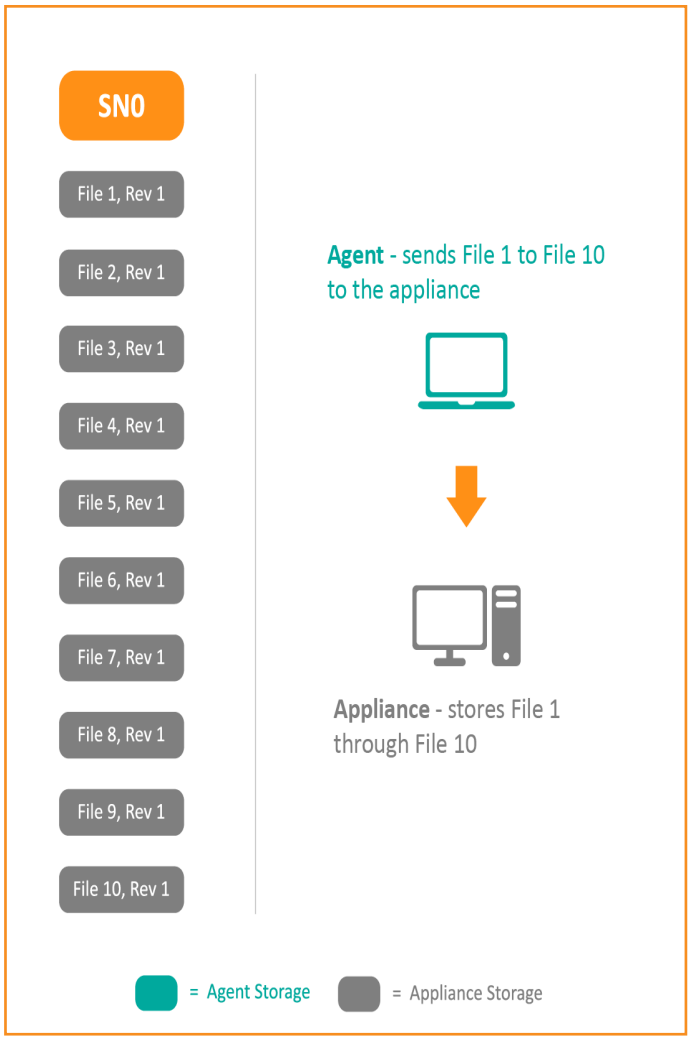
Although appliances are usually on premises machines, they can also reside in the cloud. The advantage of placing software appliances on-premises is that cloud data transfer over the internet is typically much slower than your local area network. As a result, data transfers and retrievals might not meet the Recovery Time Objective (RTO).

Web Application (aka Platform)

The web application is the overall monitoring and management system. It is where you go to monitor active archiving, set up policies, manage users, and much more. The web application does not contain any source data and is not involved in any data transfers. It is used as a powerful centralized user interface for you to manage and monitor the overall active archive process.

Storage Model

To understand how Aparavi active archive works, it is necessary to define snapshots, checkpoints, archives, and policies. Snapshots, checkpoints, and archives are all a form of data retention. Policies define the archive scheme (e.g., which files to archive, how long to retain). Default policies are available and most settings in the default polices can be used without modification and comprise best practices.



Policies

Definition

Policies specify the type of data that should be protected, the frequency, the data retention strategy, long-term storage services, and more.

Purpose

The purpose of policies is to ensure an organization-wide standard for data retention.

Snapshots

Definition

Snapshots (SN) are the basic unit of storage. Creating a snapshot is the starting point for all active archive related activities. The initial snapshot (SN0) is a complete copy of all data you choose to archive. Subsequent snapshots only contain data that has changed since the last snapshot.

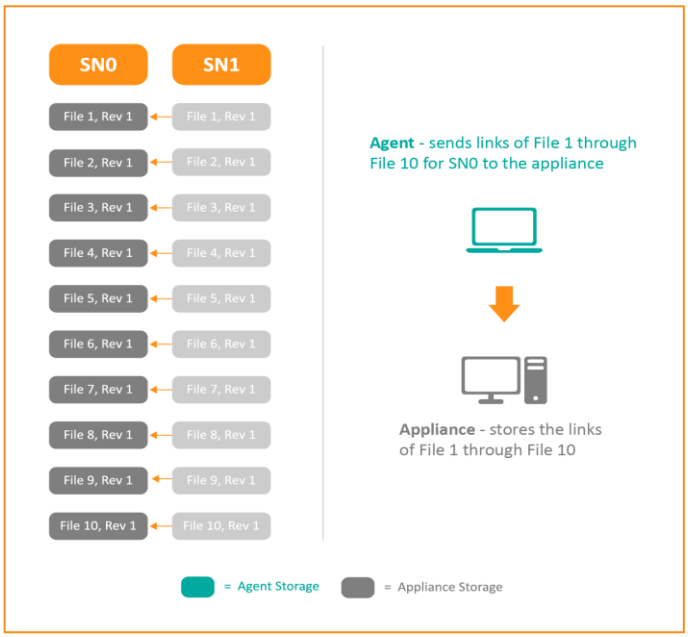
Purpose

The purpose of snapshots is to store a complete data set from a given point in time for short term recovery purposes.

Snapshots are always sent from the agent to the appliance. They never reside on the agent.

The following pages show a simple snapshot example. For this scenario, ten files are stored on the agent in a directory that the policy dictates should be archived.

Once the web application sends a valid policy to the agent, the agent starts processing the policy to create the first snapshot, which we call SN0. As mentioned above this initial snapshot is a copy of all data that complies with the defined policy - in this case, all ten files.

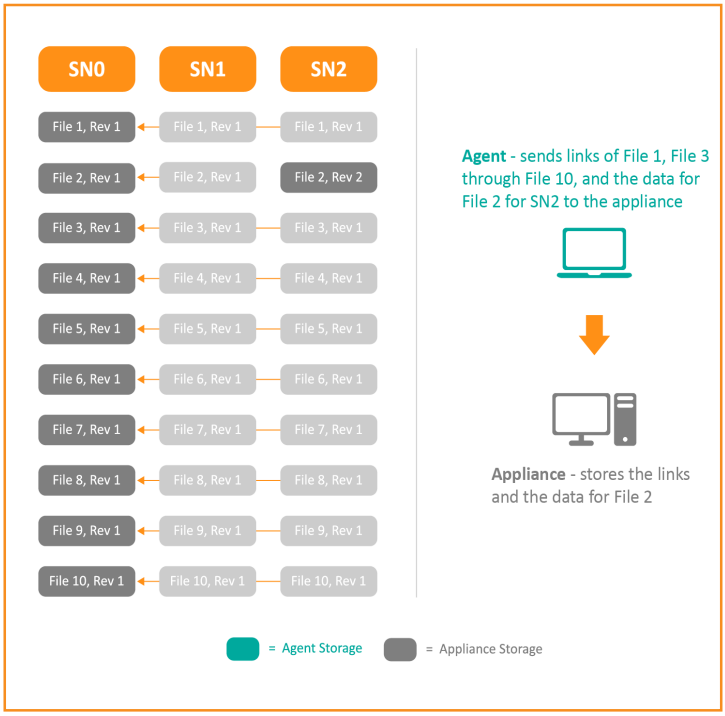


At the next scheduled snapshot (typically once per day) the next snapshot will be created - in this case, snapshot 1 (SN1). Snapshot 1 will only contain files that were changed since SN0 was created.

For this scenario let's assume none of the ten files were changed. Thus, the agent will only send links / pointers to SN0 over to the software appliance telling it that there is no need to store identical data a second time:

For the following snapshot, SN2, let's assume File 2 has changed. Again, the agent doesn't need to resend the complete set of data for Files 1 through 10, since the appliance already has those files in the exact same state. The only data the agent needs to send to the appliance is data for File 2. For all other files, the agent once again just sends links to the prior snapshot.

This process will continue in the exact same manner until you have reached your predefined snapshot retention period (typically one per day).



Checkpoints

Definition

Checkpoints (CP) are very similar to snapshots. They contain the changes made since the last snapshot instance and are stored locally on the agent until the next scheduled snapshot.

Purpose

The purpose of checkpoints is to provide a means to retrieve data easily from any desired point in time by ensuring that any changed data is protected until the next scheduled snapshot.

Checkpoints are always stored on the agent and are never sent to the software appliance.

Once the next snapshot is sent to the appliance, typically at the end of the day, the checkpoints on the agent will be removed.

For example, a person is working on a document throughout the day on Tuesday and mistakenly deletes something from the document at 3:30 PM.

While the person is working on the document the agent creates checkpoints every hour by default (or per your policy and best practices).

To further illustrate this example, let us examine what happens at each hour during this Tuesday:

8:00 AM	CP0 (changes made since the previous day's snapshot, SN3)
9:00 AM	CP1 (changes made since CP0)
10:00 AM	CP2 (changes made since CP1)
11:00 AM	CP3 (changes made since CP2)
12:00 PM	CP4 (changes made since CP3)
1:00 PM	CP5 (changes made since CP4)
2:00 PM	CP6 (changes made since CP5)
3:00 PM	CP7 (changes made since CP6)
4:00 PM	CP8 (changes made since CP7)
5:00 PM	SN4 (changes made since SN3)

At 4:15 PM the person realizes they made a mistake, they can easily go back to the 3:00 PM (or prior) version to restore the data and retrieve what they deleted. Alternatively, if they don't realize they made a mistake until 5:30 PM, they will need to restore data from the prior day's snapshot (SN3).

Archives

Definition

Archives (AR) are for long-term storage and typically stored using a cloud service, such as AWS. An archive is a complete copy of the last snapshot.

Purpose

The purpose of archives is to provide long-term storage to conform to an organization's long term data retention strategy.

Typically, archives happen weekly, but they can be set up more frequently by making changes to the policy. A common schedule would be:

- Checkpoints every hour
- Snapshots once a day
- Archives once a week on Friday

For this complete end to end scenario, let's assume Monday is the first day the agent is configured and long-term storage is configured to use a cloud storage device.

The following is how the process will play out:

Monday

Monday morning, SN0 will be sent to the software appliance
Throughout the day on Monday checkpoints will be created locally on the agent
Monday evening, SN1 will be sent to the appliance and all Monday's checkpoints will be removed

Tuesday

Throughout the day on Tuesday checkpoints will be created locally on the agent
Tuesday evening, SN2 will be sent to the appliance and all Tuesday's checkpoints will be removed

Wednesday

Throughout the day on Wednesday checkpoints will be created locally on the agent
Wednesday evening, SN3 will be sent to the appliance and all Wednesday's checkpoints will be removed

Thursday

Throughout the day on Thursday checkpoints will be created locally on the agent
Thursday evening, SN4 will be sent to the appliance and all Thursday's checkpoints will be removed

Friday

Throughout the day on Friday checkpoints will be created locally on the agent
Friday evening, SN5 will be sent to the appliance and all Friday's checkpoints will be removed
Later Friday the first archive will be created (AR0) and sent to the cloud for long-term storage. This archive will contain the latest data stored in the snapshots created throughout the week. The software appliance will navigate through snapshots 0 to 5 to locate the latest source data to build the first archive.

Then, the following week, the process will start all over. The only difference will be that on the following Friday the next archive (Archive 1) will only contain changes since the prior archive (Archive 0).

II. Only the Changes, All the Time

Definition

Aparavi's unique active archive structure ensures that only the changes are saved, all the time.

Purpose

This data structure ensures that significantly less storage space is required as compared to traditional storage models.

Files < 1MB

If a file is smaller than 1MB, it is sent in its entirety, whether it is a snapshot, checkpoint, or archive. The benefits associated with our data grooming are not worth the resources in this scenario.

Files > 1MB

If a file is larger than 1MB, dedupe technology is employed. This means what is archived is only what has been changed since the last full version of the file was saved.

Benefits of Methodology

Aparavi active archive takes the last full copy of an archived file and, in the next snapshot, only sends the changes that have been made since the last full copy.

For example, let's say that on Monday you create a PowerPoint that is 2MB in size. The first time that PowerPoint file is sent as a snapshot (SN0), the entire 2MB file will be sent because it is the first instance of the file.

On Tuesday, if you insert one slide in the PowerPoint, only that new slide will be sent in Tuesday's snapshot (SN1) because the system recognizes that most of the file is the same as the one created on Monday.

Then on Wednesday, you insert a second slide. On Wednesday's snapshot, it's going to include slide 1 and slide 2 because two slides have changed.

On Thursday, you insert a third slide. That means Thursday's snapshot will have all three new slides in it.

On Friday, no changes are made to the PowerPoint. Following best practices, because it is Friday, the first archive will be sent, and it will contain a complete presentation with slides 1, 2, and 3 embedded in it. In other words, a copy as it exists on Friday is sent, and this is the original version from Monday, plus the three new slides.

The following Monday, no changes occur.

The following Tuesday, you insert a fourth slide. So, on Tuesday, only the fourth slide will be sent in the snapshot.

On Wednesday, Thursday, Friday, of that second week, you make no changes. Then on Friday, it needs to send an archive. It will only send the fourth slide up to the cloud because with last Friday's archive, it already sent the complete presentation that included slides 1, 2, and 3.

The second Tuesday's snapshot needs slides 1, 2, 3, and 4, but the archive already has 1, 2, and 3 in it, so the only thing it needs to send up on the following Tuesday is slide 4. Aparavi sends only the changes, all the time.

Benefits of Methodology

The benefits of storing only the changes, all the time are reduced storage capacity requirements and the ability to quickly and easily retrieve data from any point in time from any device.

III. Data Pruning and Retention

Data pruning and retention are at the heart of Aparavi's active archive methodology. In very simple terms, pruning ensures that only the minimum amount of data needed to recreate a given data point is kept in storage. This means that storage space requirements are greatly reduced compared to most other methods in use today. It is the process whereby data that is no longer needed is actually removed from the software appliance and the storage, releasing the space for other uses.

Overview

For example, let us assume a case where we start with SN0, and it contains 10 files. The retention period is five days, which means five snapshots are going to be kept at any time. In this example, a daily snapshot is going to be created. Data will change throughout the week as follows:

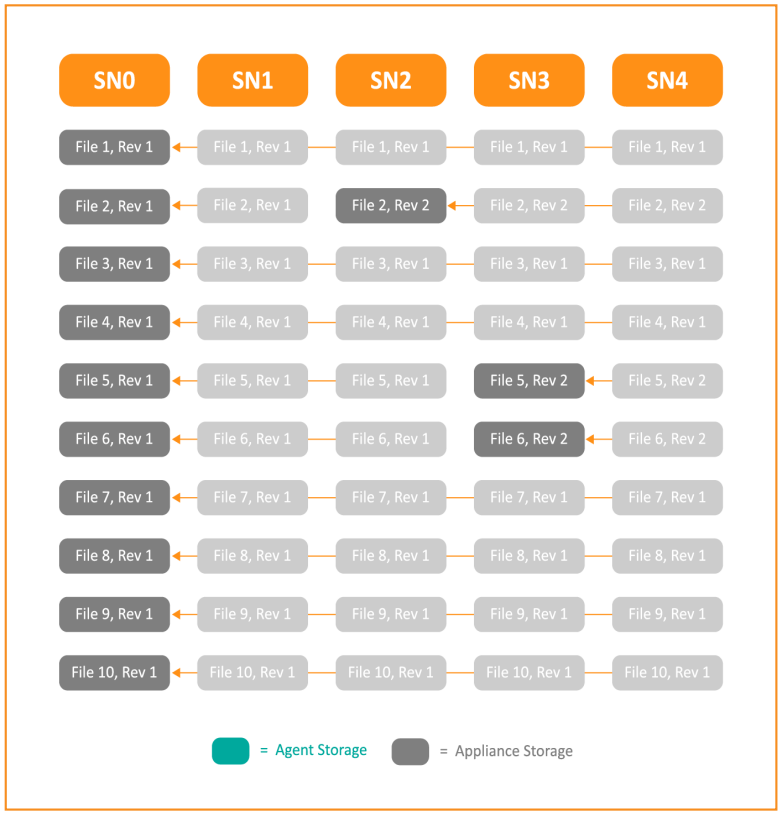
Day 1 - SN0 – contains files 1-10. The files are all marked with the revision number [Rev 1] since it is the first revision of the file that is being actively archived.)

Day 2 - SN1 – no changes – links sent to SN0

Day 3 - SN2 – contains file 2 with second revision (File 2 Rev. 2) and links back to SN0

Day 4 – SN3 – contains changes of Files 5 and 6 (F5:R2; F6:R2) and links to F2:R2 in SN2 links to SN0 for unchanged data.

Day 5 – SN4 references, SN2 for F2:R2 and SN3 for F5:R2 and F6:R2 and all the way back to SN0 for all other unchanged data.



On day 6, we have hit our retention period and can prune. The first revision of file 2 is no longer needed from SN0, so we remove it; what we have recoverable then is:

SN1 from day 2

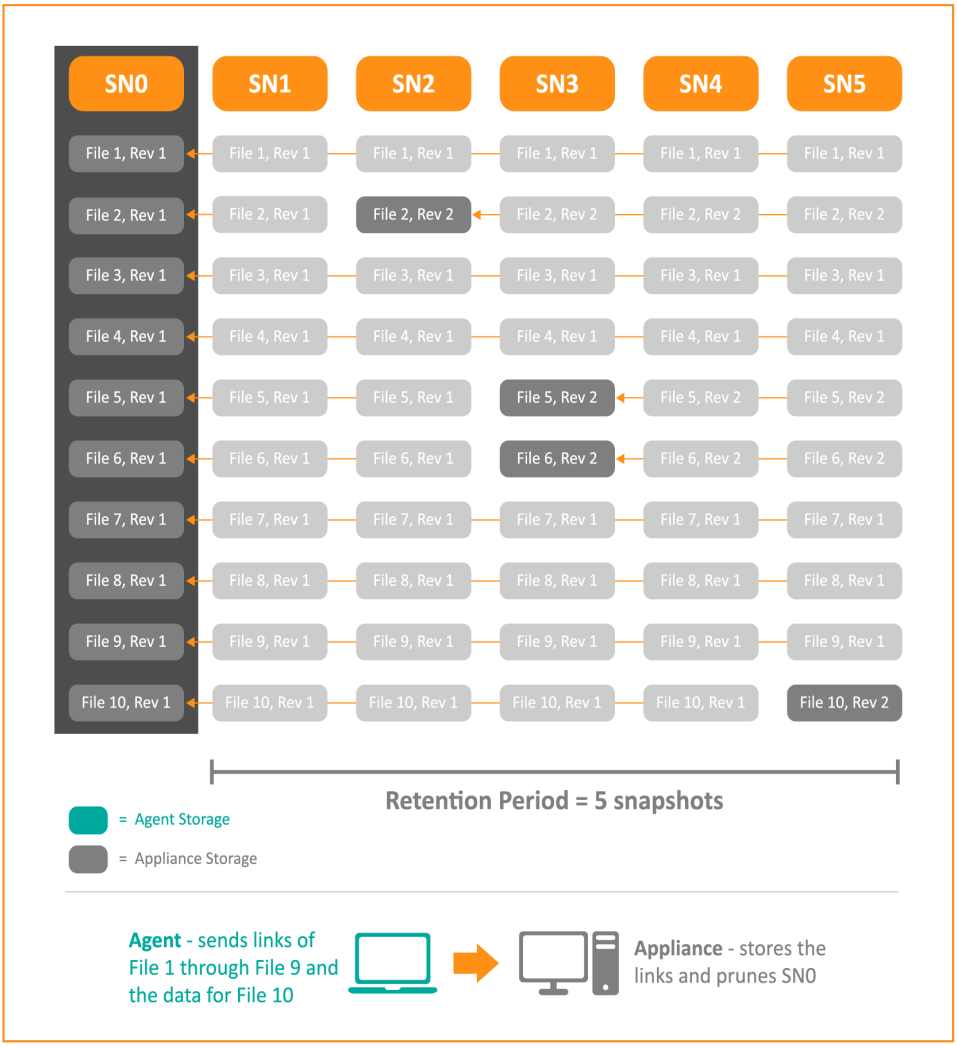
SN2 from day 3

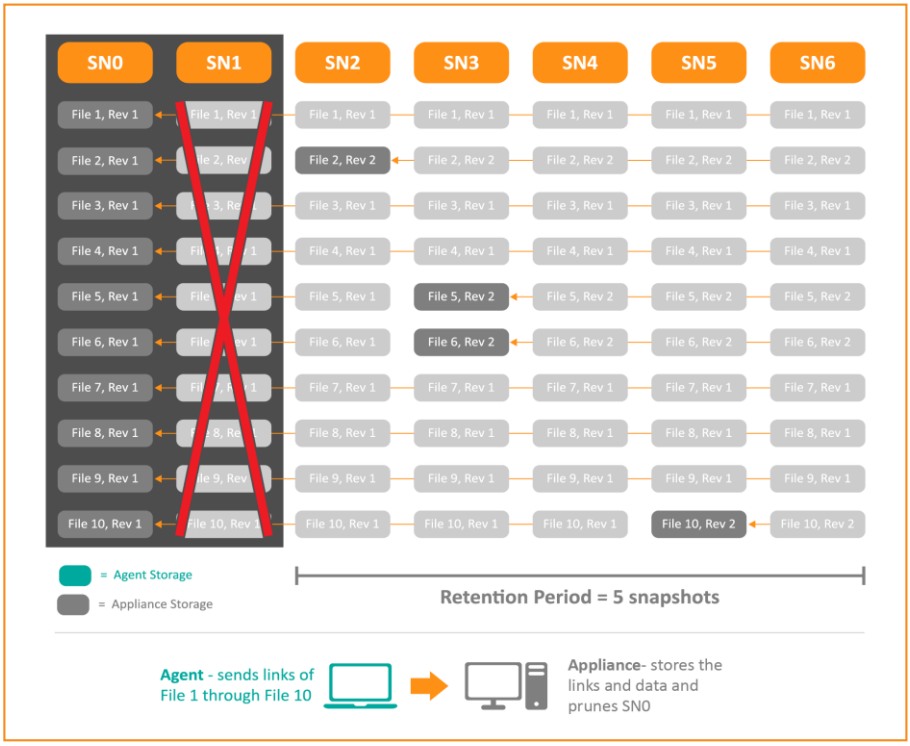
SN3 from day 4

SN4 from day 5

SN5 from day 6

Pruning means we remove all the data out of SN0 that is no longer required to support the snapshots that follow it. SN0 is not presented as a recoverable snapshot on the recover tab of the web monitor. However, even though it is not listed as one of the five recoverable files, some of the data contained in SN0 still needs to be kept. In the case of SN0, most of the data needs to be kept because one of our still recoverable snapshots (SN1) still links to data in SN0.

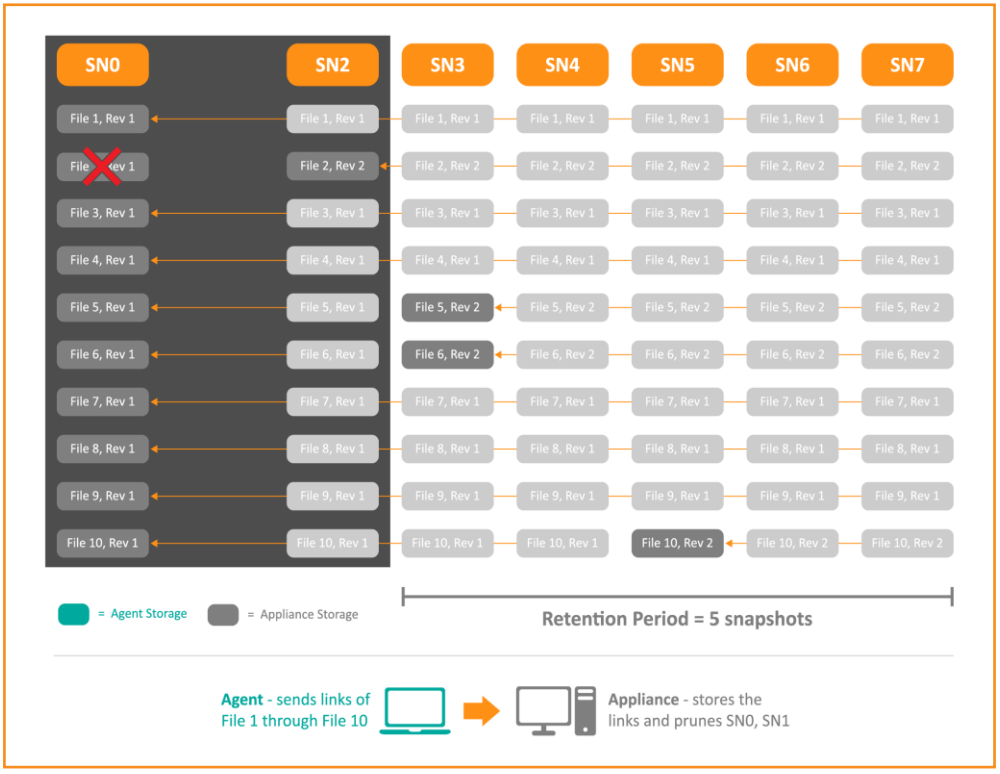




In SN6, (day 7) nothing changes and therefore the pruning process determines that there is absolutely nothing in SN1 that is needed, thus it completely removes SN1. However, all 10 files are still required from SN0, so it leaves them as they are. Even though SN0 does not show up on the recovery page, all its data is still there.

For the sake of the exercise again, no changes happen on day 8 either (SN7), so, yet again, only links are sent from the agent to the software appliance. However, something different happens during the pruning. Since File 2, Rev 1 is no longer referenced in any of our retained snapshots (SN3 through SN7), we can remove the data on the appliance occupied by File 2, Rev 1. Thus, the only recoverable revision of File 2 is now Rev2.

From this, you can see Aparavi only keeps data if it needs it to maintain your data retention policy. Instead of just continually growing your storage over time, this prunes data as it 'falls' off the retention cycle.



Pruning Archive Sets

The same process occurs with archive sets. Aparavi can be used with a rotational snapshot scheme that allows you to prune data in the middle of an archive set. That means you can decide how many daily archives you want, and how many weekly, monthly, and yearly archives you want. In this scenario, archives would be created in the following manner:

Every day – daily archive (DAR)

Every Friday – weekly archive (WAR)

Last day of every month – monthly archive (MAR)

Last day of every year – yearly archive (YAR)

Examining this from the daily level, if we began creating archives on a Monday, we would have daily archives every day, and a weekly archive on Fridays:

Monday – DAR0

Tuesday – DAR1

Wednesday – DAR2

Thursday – DAR3

Friday – WAR1

Monday – DAR4

Then, on the second Tuesday, DAR5 would be created, and this would become our 6th daily archive. We only need 5 for our retention period, so we can prune DAR0, which was the 1st daily archive.

Aparavi keeps the data we need moving forward and removes the data we no longer need that cannot be restored because it is no longer referenced.

Similarly, the retention periods for weekly, monthly, and yearly sets will determine when to remove files that are no longer needed.

For example, if the weekly retention period is set to 8, when the 9th weekly archive is created, the first weekly archive will be removed because it is no longer needed. Similarly, if the monthly retention period is set to 12, once the 13th monthly archive is created, the first monthly archive can be pruned.

Benefits of Methodology

Aparavi's storage model allows you to seamlessly recover files from any combination of local and onsite or cloud storage. Duplication is prevented by pruning data that is no longer referenced, and storage space is kept to a minimum, which saves resources.

Conclusion

Aparavi active archive helps organizations master out of control unstructured data growth. With a three-tiered architecture consisting of an Aparavi hosted platform, software appliance, and agents, we simplify the storage and archive process. The on-premises software appliance executes the policies and acts as a storage repository for file-based snapshots. It is extremely flexible, allowing you to select the number of on-premises snapshots to hold before sending data to your choice of cloud provider. As part of the active archive process, for individual file recovery, checkpoints on the server itself are used and are automatically deleted once a snapshot is verified on the appliance. A full archive is stored at designated times (e.g, end of the week). In addition to meeting corporate governance requirements, Aparavi active archive also delivers true storage independence with on-premises and multi-cloud mobility.