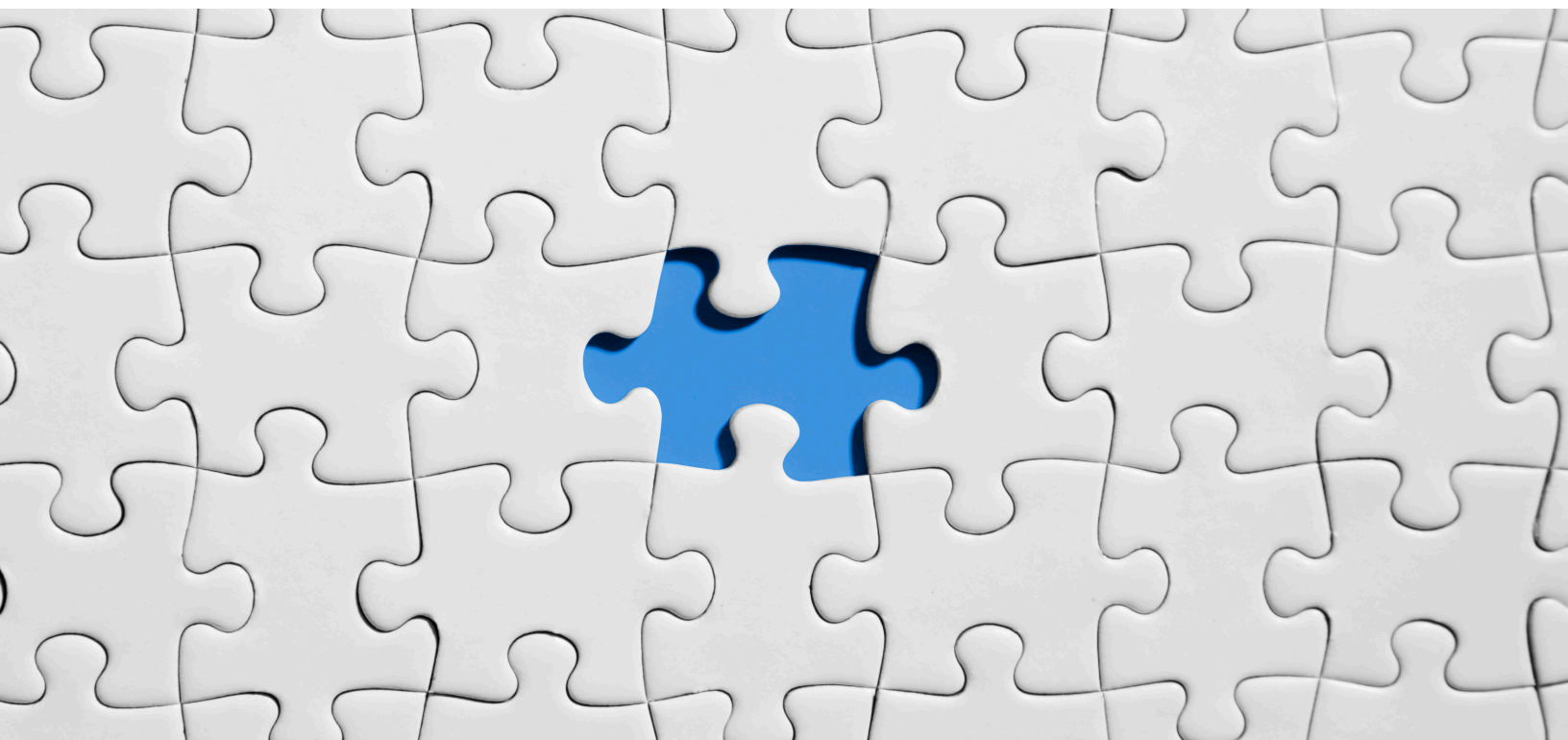


IMMUTABLE DATA PROTECTION FOR ANY APPLICATION

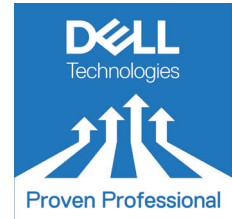


Mike van der Steen

Principal Systems Engineer

Dell Technologies

Mike.vandersteen@dell.com



The Dell Technologies Proven Professional Certification program validates a wide range of skills and competencies across multiple technologies and products.

From Associate, entry-level courses to Expert-level, experience-based exams, all professionals in or looking to begin a career in IT benefit from industry-leading training and certification paths from one of the world's most trusted technology partners.

Proven Professional certifications include:

- Cloud
- Converged/Hyperconverged Infrastructure
- Data Protection
- Data Science
- Networking
- Security
- Servers
- Storage
- Enterprise Architect

Courses are offered to meet different learning styles and schedules, including self-paced On Demand, remote-based Virtual Instructor-Led and in-person Classrooms.

Whether you are an experienced IT professional or just getting started, Dell Technologies Proven Professional certifications are designed to clearly signal proficiency to colleagues and employers.

[Learn more at www.dell.com/certification](http://www.dell.com/certification)

Table of Contents

Introduction	4
Protecting Data from Cyber-Attacks	4
Available Solutions	5
Protecting Any Application Data Beyond the Current Solutions	5
Dell Technologies PowerProtect DD Series Appliance	5
Logical Layout with MTrees	5
Immutability with Retention Lock	6
Enforcing Retention Lock on Data	7
Independent Copies with Fast Copy	8
Solution Overview of Immutable Data Protection for Any Application	9
Creating Immutable Data Copies	9
Recovery of Protected Data	13
Scripting the Solution for Automatic Protection	16
Configure the Private/Public Key for Client-Side Scripting	16
Sample Shell Script to Create and Remove Fast Copy Instances	17
Scheduling the Script to Run Automatically	19
Immutable Data Protection Example for Avamar	20
Solution Requirements	20
Protecting Avamar Backups	22
Recovering Avamar and Protected Data	24
Summary	35
References	35

Disclaimer: The views, processes or methodologies published in this article are those of the author. They do not necessarily reflect Dell Technologies' views, processes or methodologies.

Introduction

Most organizations have Data Protection as a top priority to ensure business continuity and this is even more relevant today with the prevalence of Cyber-attacks. While these Cyber-attacks traditionally originate from a source external to the organization, attacks from bad actors internal to the organization is unfortunately a real threat. Mitigation strategies to protect the organization's data includes hardening of operating systems, applications and devices. Additionally, regular backups must be taken of the data to provide a level of protection, but these do not safeguard data when the threat attacks both the backup systems and the data itself.

Dell Technologies PowerProtect DD Series appliances³ (formally known as Dell EMC Data Domain and will be referred to as Data Domain in this article) provide Retention Lock capability. This feature, when enabled and applied to data stored on the appliance, prevents that data from being deleted or modified in any way. This results in backup data being stored on the appliance in an immutable manner.

This Knowledge Sharing article provides an overview of the fundamental Data Domain features leveraged to create immutable copies of backup data, the process of creating them, the recovery process and a sample script to automate the daily tasks of creating these immutable copies. An example is provided for creating immutable copies of backups taken with Dell Technologies Avamar and the recovery process. Avamar is used as an example as this backup application at the time of writing this article does not have the capability to natively integrate with Data Domain Retention Lock feature.

While the primary focus of this article is based on backup applications, it can be extended to incorporate any application storing data on Data Domain, hence providing immutable data protection for any application.

Protecting Data from Cyber-Attacks

There are several strategies that can be implemented to protect the organizations data from Cyber-attacks or internal bad actors. Dell Technologies recognizes the need to safeguard data and discuss the Good, Better, and Best in a Layered Cyber Security for Data Protection approach as shown in **Figure 1**.

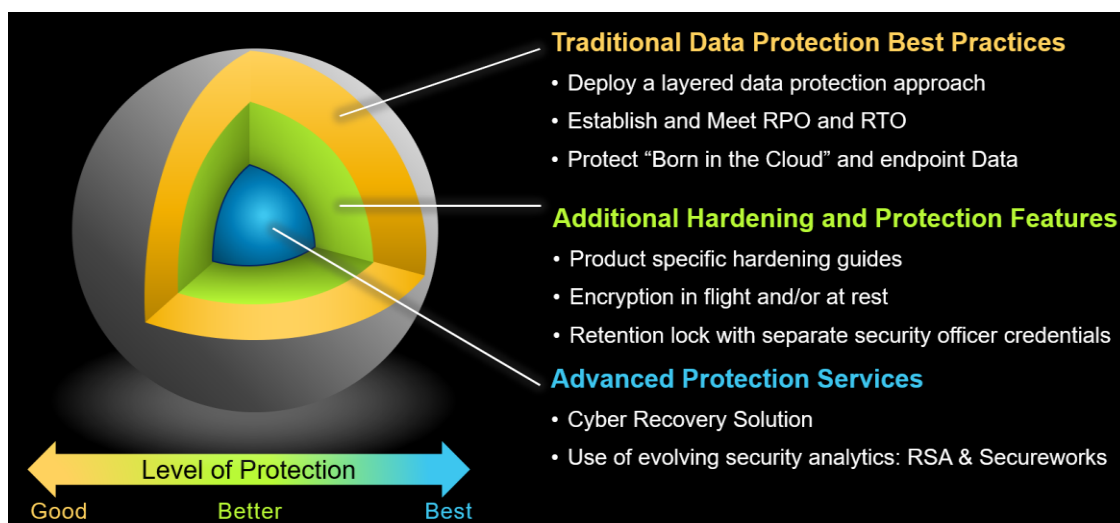


Figure 1- Dell Technologies layered Cyber security for data protection¹

The base layer or a good level of protection is achieved by performing regular backups of the organization's data. The second layer or better level of data protection requires hardening of the backup application and appliances. In addition, leveraging retention lock capability to ensure data is stored in an immutable manner. Finally, the last layer or best level of data protection requires a cyber recovery solution where the organization's most valuable data is stored in a separate or air-gapped environment.

This solution focuses on providing a better level of data protection for those applications that do not integrate natively with Data Domain Retention Lock and enabling protected data to be stored in an immutable manner.

Available Solutions

Dell Technologies' data protection portfolio includes a Cyber Recovery Solution² where data is safeguarded in an isolated environment designed to protect the organization's most critical applications. This type of data normally represents a small percentage of the overall data managed by the organization's Information Technology (IT) department.

Organizations that use NetWorker⁴ and/or PowerProtect Data Manager⁵ have native integration with Data Domain Retention Lock feature. This enables those organizations to achieve a better level of data protection.

Protecting Any Application Data Beyond the Current Solutions

There are numerous data protection applications in the market. Some applications, including Dell Technologies Avamar are unable to leverage Data Domain Retention Lock natively, which means that data protected by Avamar is not stored in an immutable state.

To help organizations achieve a better level of data protection by storing protection data in an immutable state, a scripted based solution is provided in this article. While the primary focus of this solution was designed for organizations using Avamar, it can in fact be applied to any application writing data to a Data Domain appliance.

Dell Technologies PowerProtect DD Series Appliance

At the core of the solution described in this article and of any Dell Technologies Data Protection solution is the PowerProtect DD Series appliance, otherwise known as Data Domain. Data Domain is not exclusive to Dell Technologies Data Protection solutions and is used with many industry-leading data protection software applications. There are numerous key features which make it a perfect protection storage platform for organizations, including its variable length deduplication algorithm, global deduplication within each appliance and the Data Invulnerability Architecture to name just a few.

Features relevant to the solution including MTrees, Retention Lock and Fast Copy, are explored in detail. Additional information about these and other features of Data Domain can be found in the Data Domain Operating System Administration Guide⁸ and Command Reference Guide⁹.

Logical Layout with MTrees

When a Data Domain appliance is initially configured, it has a single file system and MTrees are created to provide logical partitions of that single file system. An MTree is a directory in its simplest form, however it allows for granular operations to be performed on the MTree, which include defining security access, quotas, replication, Retention Lock and creating snapshots.

When applications are configured to write data to Data Domain, they are directed to use a predefined MTree. The MTree can be exposed to the application via Virtual Tape Library (VTL), Common Internet File System (CIFS), Network File System (NFS) or as a Data Domain Boost device. For future reference, when a Data Domain Boost (DDBoost) device is created, it is referred to as a Logical Storage Unit (LSU) within the Dell Technologies documentation and this is linked to an MTree on the Data Domain appliance.

Figure 2 shows a list of MTrees that exist on Data Domain appliance called DD01. The first three MTrees are assigned to Avamar applications, while the last two MTrees are used by Dell Technologies PowerProtect Data Manager. For reference, the backup MTree is created by default at the time the Data Domain File System is initialized and normally not used by applications.

```

sysadmin@dd01# mtree list
Name                               Pre-Comp (GiB)  Status
-----
/data/col1/avamar-1577926088        42.6            RW
/data/col1/avamar-1577931553        33.1            RW
/data/col1/avamar-1577945908        42.6            RW
/data/col1/backup                    0.0            RW
/data/col1/Default                   0.0            RW
/data/col1/pp01-dr                   1.1            RW
/data/col1/VM_Images_-_DR-pp01-af1ac 7305.0          RW/RLGE
-----
D   : Deleted
Q   : Quota Defined
RO  : Read Only
RW  : Read Write
RD  : Replication Destination
RLGE : Retention-Lock Governance Enabled
RLGD : Retention-Lock Governance Disabled
RLCE : Retention-Lock Compliance Enabled

```

Figure 2 - List of MTrees on Data Domain appliance DD01

When individual credentials are set for every MTree or LSU, it provides an additional layer of security to ensure data written by one application cannot be accessed by another.

Immutability with Retention Lock

The Retention Lock feature was introduced many years ago and is included as a standard feature with any Data Domain appliances running DD OS 6.x or greater. Retention Lock can be applied to a MTree either as Governance or Compliance mode, but not both. With Retention Lock applied to data, that data cannot be modified, overwritten or deleted for the set period as defined by the Retention Lock settings.

There are two Retention Lock options available on the Data Domain appliance; Governance and Compliance modes. In Governance mode, data is retained for a specific time period that aligns with the organization's internal IT governance policy and implemented by the system administrator. Simply put, with Governance mode, the system administrator is trusted.

Compliance mode is used when the organization needs to adhere to strict regulatory standards, like those provided in the SEC 17a-4(f)⁶. Where possible, the recommended mode of Retention Lock is Compliance

mode as the Data Domain sysadmin account can use override commands that can modify or delete data locked with Governance mode.

Please note that the physical Data Domain appliance supports both Governance and Compliance modes of Retention Lock. The Data Domain Virtual Edition (DDVE) only supports the Governance Retention Lock mode.

Looking back to Figure 2, the status of the last MTree that is used by PowerProtect Data Manager is shown as RW/RLGE. This indicates that the MTree is configured with Retention-Lock Governance Enabled.

Enforcing Retention Lock on Data

When Data Domain Retention Lock was first introduced, the application writing data to the appliance was required to 'touch' the file by modifying the access time (atime)⁷. With the atime set to a date and time in the future, Data Domain Retention Lock will adhere to this atime and prevent modification or deletion of the data until the atime has passed. Using this method allows for an application to set the atime for each file individually, which provides great flexibility and control.

To configure Data Domain Retention Lock, enable it on the desired MTree with the manual mode option as shown in **Figure 3**.

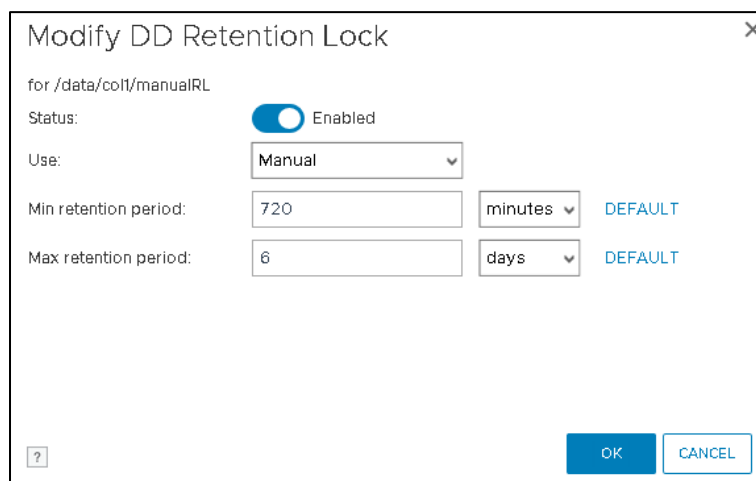


Figure 3 - Data Domain manual Retention Lock feature settings

When enabling this feature, there are two parameters which need to be defined for Retention Lock to be enabled. These parameters include;

- **Min retention period** – defines the minimum retention period that will be applied to the data and 720 minutes (12 hours) is the minimum. The default setting is 720 minutes.
- **Max retention period** – defines the maximum retention period that can be applied to the data. The default setting is 1827 days.

An alternative option to touching a file is to leverage Data Domain's Automatic Retention Lock feature released with Data Domain Operating System version 6.2.0.30. This feature applies only to MTrees exposed to applications via CIFS and NFS. In **Figure 4**, the Automatic Retention Lock feature has been enabled on the desired MTree.

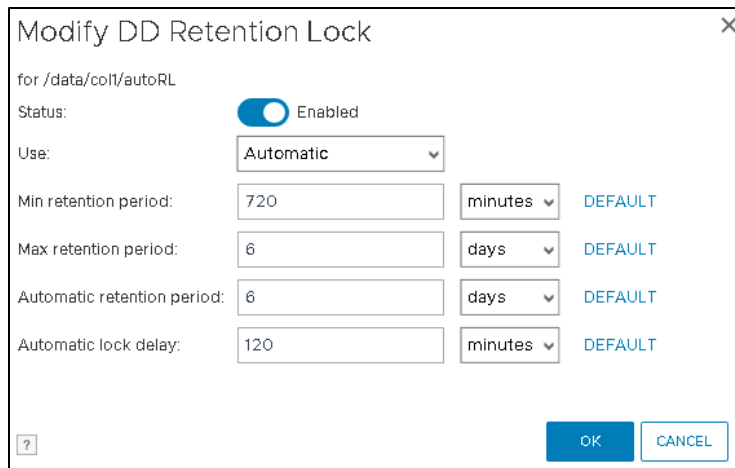


Figure 4 - Data Domain automatic Retention Lock feature settings

When enabling this feature, there are several parameters which need to be defined for Retention Lock to be enforced. These parameters include;

- **Min retention period** – defines the minimum retention period that will be applied to the data and 720 minutes (12 hours) is the minimum. The default setting is 720 minutes.
- **Max retention period** – defines the maximum retention period that can be applied to the data. The default setting is 1827 days.
- **Automatic retention period** – defines the retention period that is applied to the data and the value must be greater than the minimum, but not exceed the maximum retention period. The default value is 720 minutes.
- **Automatic lock delay** – defines the time period after which a file was modified before the retention lock time period is applied. The minimum value that this can be set to is 5 minutes and the default value is 120 minutes.

In summary, the values defined in **Figure 4** will apply Retention Lock to the data in the desired MTree for a period of six days once the file has not been modified for 120 minutes.

With Data Domain Automatic Retention Lock feature, data can be stored in an immutable manner for the duration defined by the value set in the Automatic retention period section. There is no granular control to specify different retention periods to data stored within the MTree.

The solution described in this article will leverage Data Domain Automatic Retention Lock feature.

Independent Copies with Fast Copy

The Data Domain Fast Copy feature is a process that creates a copy of files or a directory tree from a defined source directory to a defined target within the same appliance. Note that the Fast Copy process cannot be used to create a copy of data between Data Domain appliances. It is similar to taking a snapshot, however, Fast Copy feature is extremely efficient as it is a pointer-based copy of metadata associated with the data and does not physically duplicate the data on the Data Domain appliance.

The process of creating a Fast Copy of a file or directory tree is performed via a command line operation. Examples are provided in the Solution Overview section.

Solution Overview of Immutable Data Protection for Any Application

The solution leverages Data Domain Fast Copy to create a copy of the MTree used by the application to a different MTree. Within this MTree, Data Domain Automatic Retention Lock feature is applied to ensure data 'copied' to this MTree cannot be deleted for the duration defined in the Retention Period parameter. An immutable copy of data now exists independently from the original MTree used by the application as shown in **Figure 5**.

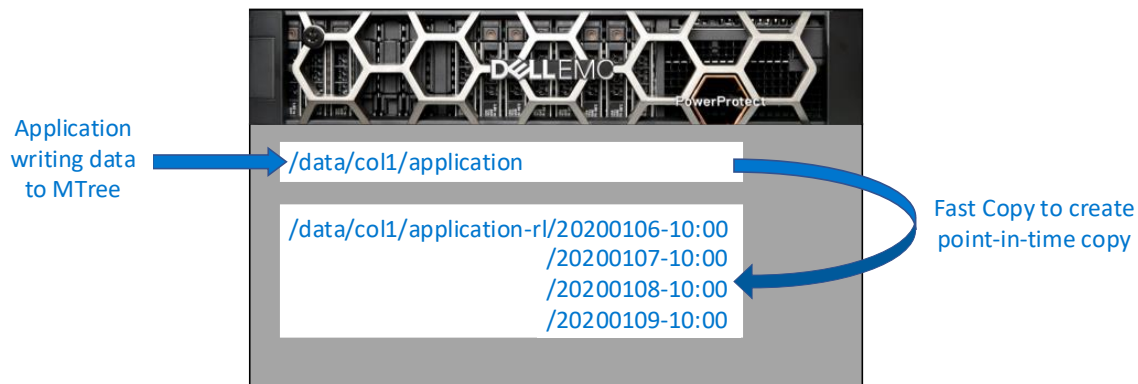


Figure 5 - Solution overview with Fast Copy operations to a 'Retention Lock' MTree

Depending on the application used, this 'Retention Lock' MTree not only stores data, but may also store meta or configuration data critical to the operational integrity of that application.

Having a copy of the data in an immutable manner is required to achieve a better level of data protection. Even more critical to the organization is the recovery of that data in the event of a Cyber-related or inside bad actor attack. The first step in recovering the data is to use the Data Domain Fast Copy command to make a copy of the data stored in the 'Retention Lock' MTree to a recovery MTree. From here, recovery of the backup application and the data can be performed.

For example, an application like Avamar cannot natively use Retention Lock, nor can the MTree that it writes to have Retention Lock enabled. Avamar can be configured to save its checkpoint to the Data Domain MTree where the clients it protects have their backup data stored. The Avamar checkpoint contains all relevant data to recover Avamar in the event of a disaster. So, if Avamar is compromised or subject to a Cyber-attack, Avamar can be fully recovered from a Fast Copy stored in the 'Retention Lock' MTree. A complete step by step example of this process is provided later in the article.

Before diving into the detailed process and commands used to enable the creation of immutable data copies, let's review the process at a higher level.

1. Identify the MTree used by the application
2. Create a 'Retention Lock' MTree and enable automatic Retention Lock
3. Run Fast Copy commands to create point-in-time instances of the application MTree to the 'Retention Lock' MTree

Creating Immutable Data Copies

To create immutable copies of data from an application writing to an MTree on Data Domain, review the steps outlined below. The following process assumes that the application has already been configured to write to the Data Domain appliance.

Step 1 – Identify the MTree used by the application

Log on to the Data Domain System Manager using a web browser or via an SSH connection to the Data Domain appliance.

Review the list of existing MTrees defined in the Data Domain appliance and identify the MTree associated with the application. Take note of the MTree name as it will be used in the following step.

For example, using an SSH connection to the Data Domain appliance, run the command *mtree list* to view a complete list of all MTrees and their status as shown in Figure 6. The MTree for this example is */data/col1/application*

```
sysadmin@dd01# mtree list
Name                               Pre-Comp (GiB)  Status
-----
/data/col1/application              1.0  RW
/data/col1/avamar-1577926088        42.6  RW
/data/col1/avamar-1577931553        33.1  RW
/data/col1/avamar-1577945908        42.6  RW
/data/col1/backup                    0.0  RW
/data/col1/Default                   0.0  RW
/data/col1/pp01-dr                   1.1  RW
/data/col1/VM_Images_-_DR-pp01-af1ac 7305.0 RW/RLGE
-----
D   : Deleted
Q   : Quota Defined
RO  : Read Only
RW  : Read Write
RD  : Replication Destination
RLGE : Retention-Lock Governance Enabled
RLGD : Retention-Lock Governance Disabled
RLCE : Retention-Lock Compliance Enabled
```

Figure 6 - List of MTrees using the MTree list command

Step 2 – Create and Configure the Retention Lock MTree

Log on to the Data Domain System Manager using a web browser or via an SSH connection to the Data Domain appliance. A new MTree will be created as the 'Retention Lock' MTree, have Retention Lock enabled and an NFS export created.

For example, connecting to Data Domain appliance via SSH the command *mtree create mtree-path* will be used to create the 'Retention Lock' Mtree. In Figure 7 the *mtree create /data/col1/application-rl* command is run. The name of the MTree created is a combination of the name of the MTree used by the application and appended with *-rl*. This enables easy identification of retention lock MTrees and their association to the application data it is protecting.

```
sysadmin@dd01# mtree create /data/col1/application-rl
MTree "/data/col1/application-rl" created successfully.
Quota soft limit: none, hard limit: none(*)
(*) Quota Capacity is disabled. Capacity limits not enforced.
```

Figure 7 - Create MTree using the MTree create command

With the 'Retention Lock' MTree created, Retention Lock needs to be enabled and configured to match the organizations' requirements. Again, using an SSH connection to Data Domain, the *mtree retention-*

lock enable mode {compliance | governance} mtree mtree-path command will be run. In **Figure 8** the command *mtree retention-lock enable mode governance mtree /data/col1/application-rl* is run.

```
sysadmin@dd01# mtree retention-lock enable mode governance mtree /data/col1/application-rl
Retention-lock feature is enabled for MTree /data/col1/application-rl.
```

Figure 8 - Enable Retention Lock using the MTree retention-lock enable command

Now that Retention Lock has been enabled, the parameters need to be set to match that of the organizations' policy. Continuing to use SSH, the *mtree retention-lock set {min-retention-period | max-retention-period | automatic-retention-period | automatic-lock-delay} period mtree mtree-path* command will be run.

In Figure 9 the following commands were run to set each parameter for Retention Lock. For this example, the same parameters as defined in Figure 4 are specified in the commands.

```
mtree retention-lock set min-retention-period 720min mtree /data/col1/application-rl
mtree retention-lock set max-retention-period 6day mtree /data/col1/application-rl
mtree retention-lock set automatic-retention-period 6day mtree /data/col1/application-rl
mtree retention-lock set automatic-lock-delay 120min mtree /data/col1/application-rl
```

When the *set automatic-retention-period* option is executed, the Retention Lock is changed from manual to automatic mode as seen in Figure 9.

```
sysadmin@dd01# mtree retention-lock set min-retention-period 720min mtree /data/col1/application-rl
Retention-lock min-retention-period of MTree /data/col1/application-rl is set to 720min.
sysadmin@dd01#
sysadmin@dd01# mtree retention-lock set max-retention-period 6day mtree /data/col1/application-rl
Retention-lock max-retention-period of MTree /data/col1/application-rl is set to 6day.
sysadmin@dd01#
sysadmin@dd01# mtree retention-lock set automatic-retention-period 6day mtree /data/col1/application-rl

All new files written to this MTree from now on will be Automatic Retention Locked.

Do you want to enable Automatic Retention Lock on this MTree? (yes|no) [no]: yes
Retention-lock automatic-retention-period of MTree /data/col1/application-rl is set to 6day.
sysadmin@dd01#
sysadmin@dd01# mtree retention-lock set automatic-lock-delay 120min mtree /data/col1/application-rl
Retention-lock automatic-lock-delay of MTree /data/col1/application-rl is set to 120min.
```

Figure 9 - Setting the Retention Lock parameters via the mtree retention-lock set command

With Retention Lock enabled and configured, the final step is to create an NFS export of the 'Retention Lock' MTree. This is required so that Fast Copy copies can be removed once the retention period has been met. If this is not done, expired data ready for cleaning may not be removed from the Data Domain appliance. For example, the application may have marked data ready for cleaning from Data Domain, however, if a Fast Copy of that data still exists in the 'Retention Lock' MTree, it will not be removed until the Fast Copy is removed. Data Domain will never delete data from the file system until there are zero metadata pointers linked to a unique data segment. This is a key attribute and feature of Data Domain's Data Invulnerability Architecture.

With that in mind, an NFS export needs to be created, and like all previous steps, a SSH connection will be used to create the NFS export by running the `nfs export create [export-name] path [clients client-list [options option-list]] [referral referral-name remote-servers address-list [remote-path path]]` command.

For this example, the `nfs export create application-rl path /data/col1/application-rl clients *.mlab.internal` command was run via an SSH connection as shown in **Figure 10**. Depending on the organization's IT security policy, optional parameters for the `nfs export create` command may be required.

```
sysadmin@dd01# nfs export create application-rl path /data/col1/application-rl clients *.mlab.internal
NFS export 'application-rl' created.
```

Figure 10 - Creating the NFS export via the NFS export create command

Step 3 – Create Fast Copy of the Application MTree

Knowing the MTree of the application and having completed the configuration of the 'Retention Lock' MTree, Fast Copy operations can be performed to provide immutable data protection copies. The process is for the Fast Copy command to select the entire application MTree and create a 'copy' to a directory within the 'Retention Lock' MTree. The directories created for each Fast Copy instance need to have a unique name, and one recommendation is to use a date/time format for each Fast Copy operation.

The format that is used in the below examples uses the current date and time of the Fast Copy creation operation in the format of `yyyymmdd-hh:mm`. In the below example, an SSH connection was made to the Data Domain appliance and a Fast Copy command was run twice, once at 2:06pm and again at 3:06pm on 6th January 2020 with the source directory of the application MTree as shown in Figure 11.

```
sysadmin@dd01# filesys fastcopy source /data/col1/application destination /data/col1/application-
rl/20200106-14:06
(00:00) Waiting for fastcopy to complete...
Fastcopy status: fastcopy /data/col1/application to /data/col1/application-rl/20200106-14:06: copied 93 files,
12 directories in 0.09 seconds
sysadmin@dd01#
sysadmin@dd01# filesys fastcopy source /data/col1/application destination /data/col1/application-
rl/20200106-15:06
Fastcopy status: fastcopy /data/col1/application to /data/col1/application-rl/20200106-15:06: copied 93 files,
12 directories in 0.06 seconds
```

Figure 11 - Fast Copy command run against the application MTree

This Fast Copy command was run manually, however, it is recommended to have it scheduled to run once or twice a day, depending on the organization's requirements and the application data to be protected. A sample Shell script is provided on page 16 to enable automatic data protection for the desired application MTree and was tested on a Centos 7 server.

One requirement of the automation script is for the 'Retention Lock' MTree to be exported via NFS and mounted on a Linux server. Mounting the NFS export on the Linux server enables data to be cleaned/removed once it has met the retention period and enables completed Fast Copy operations to be viewed.

To confirm that the Fast Copy operations were successful, the NFS export of the 'Retention Lock' MTree is mounted and the contents listed as shown in Figure 12 below.

```
[root@linux10 /]# mount -t nfs -o hard,intr,nolock,nfsvers=3,tcp,rsize=1048600,wsz=1048600,bg
dd01.mlab.internal:/data/col1/application-rl /mnt/nfs/application-rl
```

```
[root@linux10 /]#  
[root@linux10 /]# ls -l /mnt/nfs/application-rl/  
total 1  
drwxrwxrwx. 3 root root 159 Jan 3 15:24 20200106-14:06  
drwxrwxrwx. 3 root root 159 Jan 3 15:24 20200106-15:06
```

Figure 12 - CentOS mount command and listing contents of 'Retention Lock' MTree

With the "Retention Lock" MTree export mounted, a list command was run to view the contents of the export. The directories 20200106-14:06 and 20200106-15:06 are listed as expected and confirms that the Fast Copy command as per Figure 11 was successful.

The retention of Fast Copy directories from 'Retention Lock' MTree and when they are performed during the day needs to be given due consideration.

First, let's look at when the Fast Copy operations should be performed. It will be governed by the application and by the organization's IT policies. To ensure that the data processed by the application can be recovered, it is recommended to create a Fast Copy of the application MTree when the application is in a known state. This could mean while the application is 'offline' for any scheduled maintenance or following any maintenance activities. Consult the Subject Matter Expert (SME) of the application to confirm the best time for the Fast Copy operation to be performed.

The second aspect relates to the time that the Fast Copy instances should be retained for. This is determined by the retention period set on the 'Retention Lock' MTree. Keep in mind that a Fast Copy instance created in an MTree that has automatic Retention Lock enabled, cannot be removed until the automatic retention period parameter has expired. That is, if the automatic retention period parameter is set to 14 days, then a Fast Copy created in that MTree cannot be modified or deleted during that 14 day period. On the 15th day, the Fast Copy instance has exceeded the 14 days retention period and can then be deleted.

Recovery of Protected Data

The primary purpose of data protection is to recover data if the original data has been deleted, become corrupted or simply unavailable. This section discusses two recovery options. However, the preferred recovery option will depend on the application itself.

Before proceeding to the available recovery options from the Fast Copy instances, it is important to consider when to use these recovery options. For day to day data recovery operations, this should be provided through the application itself.

The recovery described in this section provides the organization recovery points-in-time, in case the application's data has been corrupted or deleted via malicious acts. The extent of the recovery depends on the application and its configuration. Metadata or catalog files can also be recovered to the same point-in-time. Ideally, these critical files or metadata, along with the data itself are all stored in the MTree used by the application.

Before diving into the each of the recovery options available, let's review the process at a higher level.

1. Determine if the application to be recovered requires the same MTree name or if it can use an alternative MTree
2. Perform a Fast Copy operation of the application MTree as a rollback point; this is especially important if data is to be recovered to the application MTree itself

3. Perform the Fast Copy operation of the point-in-time instance that needs to be recovered from
4. Recover the application and data

Option 1 – Recovery to the Original MTree

Use this option if the application requires that the name of the MTree on the Data Domain be the same. A list of applications that have this requirement is not listed in this article. Consult your application SME to determine if this is a requirement.

With this recovery option, a point-in-time Fast Copy stored in the ‘Retention Lock’ MTree is copied to the original MTree. During this operation, all contents stored in the original application MTree will be overwritten. It is recommended to shut down the application and then perform a manual Fast Copy operation of the application MTree to the ‘Retention Lock’ MTree just prior to the recovery operation. This provides an application-consistent recovery point should this be a requirement.

An overview of the recovery process whereby the original MTree is overwritten is shown in Figure 13.

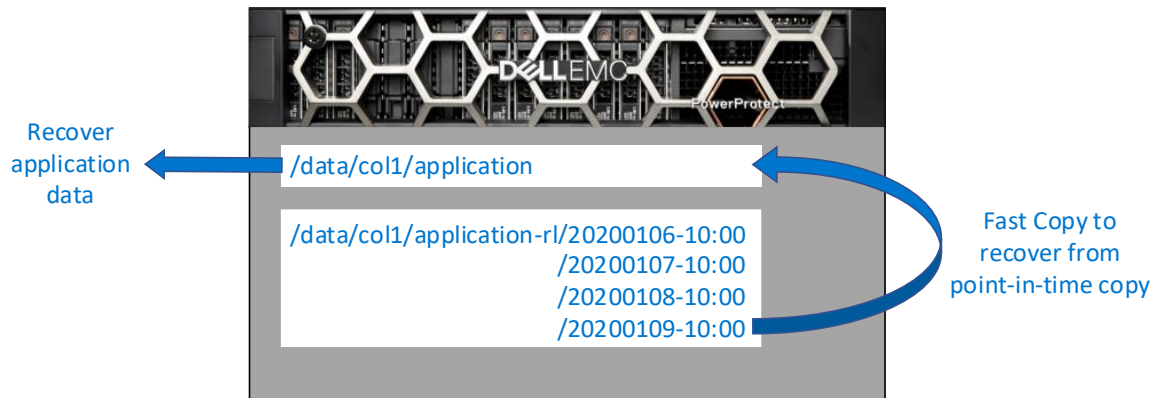


Figure 13 – Option 1 recovery whereby the original application MTree is overwritten

Once the data from the point-in-time has been copied to the application MTree via Fast Copy, the application owner can then commence the recovery of the data. The recovery of the data is dependent on the application itself. Consult an SME for that application for detailed recovery procedure.

Option 2 – Recovery to Different MTree

If the application does not require the name of the MTree to be the same, then recover the data to a ‘Recovery’ MTree. In doing so, the contents of the original MTree are not overwritten and data can be recovered independently. Again, like the first option, the process of recovery is dependent on the application itself and it is recommended to consult an application SME first. It is recommended to shut down the application and then perform a manual Fast Copy operation of the application MTree to the ‘Retention Lock’ MTree just prior to the recovery operation. This provides an application-consistent recovery point should this be a requirement.

An overview of the recovery process is shown in **Figure 14**.

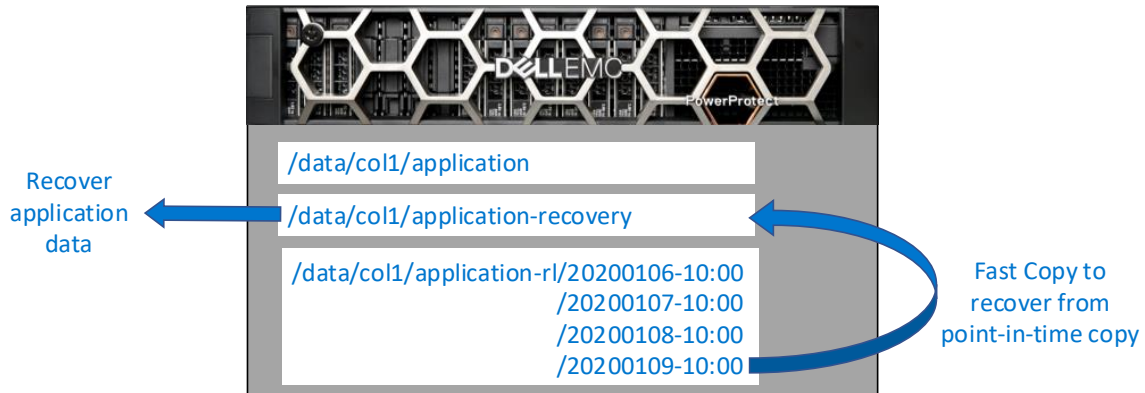


Figure 14 - Option 2 recovery where a 'Recovery' MTree is used

Once the data from the point-in-time instance has been copied to the application MTree via Fast Copy, the application owner can then commence recovery of the data. The recovery of the data is dependent on the application itself. Consult an SME for that application for the detailed recovery procedure.

Data Recovered – What Next?

Performing a recover of data from a Fast Copy instance is typically undertaken as a result of a serious incident. The circumstances of what caused the event and the impact to the application need to be analyzed. A decision can then be made on what happens next.

If the first recovery option was performed, then there are two options to consider. The first is to continue to use the application from the recovered Fast Copy instance. Alternatively, with data recovered, the application is then restored to the manually created Fast Copy instance made just prior to the recovery operation.

If the application allows for recovery to a different MTree, as described in recovery option 2, then once the relevant data is recovered, the recovery MTree may be deleted and the application continues to operate using the original MTree.

Which instance of the application continues to be used will depend on the health and integrity of the application as a result of the incident.

Scripting the Solution for Automatic Protection

The Shell script that is provided in this section is done so as a guide on how the Fast Copy operations can be automatically created and removed once the retention period has been met. Feel free to take the contents of the script and modify or enhance it as required.

The script ran successfully using a CentOS 7 Linux server hosted on VMware ESXi 6.7 and the destination Data Domain appliance was running Operating System version 6.2.0.35. To ensure that the Shell script runs in a non-interactive mode, SSH certificates authentication was configured between the Linux server and the Data Domain appliance. A private/public key pair is configured and allows SSH login to the Data Domain appliance without the requirement to enter a password.

There are several resources available online that provide an overview on how to configure SSH connections for client-side scripting. For example, a blog article titled “Client side scripting on DataDomain”¹⁰ was referenced for this scripted example.

A lab environment was set up to create and test the scripted automation approach. For reference, dd01.mlab.internal is the Data Domain appliance and the Linux CentOS 7 server is linux10.mlab.internal.

Configure the Private/Public Key for Client-Side Scripting

Log on to the Linux server that will run the script to automate the Fast Copy operations and create an SSH key pair as shown in **Figure 15**. Please note that while the blog article uses `ssh-keygen -t dsa` command, newer versions of the Data Domain operating system require `ecdsa` option instead of `dsa`. If the `dsa` option is used, a password prompt will be presented when trying to make a ssh connection to the Data Domain appliance.

```
[root@linux10 /]# ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/root/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ecdsa.
Your public key has been saved in /root/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:jxet39BI0FVRk7FhCBxHTfG/7i7JAL9MqBQnmbvXC/4 root@linux10.mlab.internal
```

Figure 15 - Run the ssh-keygen command to create a Private/Public key pair

Navigate to the `/root/.ssh` directory where the certificates are stored and copy the contents of the public certificate in full as shown in Figure 16. For security reasons, rather than omitting the output of the command, the alphanumeric string shown in Figure 16 has been modified.

```
[root@linux10 .ssh]# ls
id_ecdsa id_ecdsa.pub
[root@linux10 .ssh]#
[root@linux10 .ssh]# cat id_ecdsa.pub
ecdsa-sha2-nistp256 AAvBE2VjZHnHLXNoYtItbmlzdHAyNTYrtAAIbmlzdHAyNTYAAABBBH2/QOSk5y3b/ntSD8bki-
XcW/JM0a0+WCMF9EcYJYvbnlvsiaXpnRZ8NHU1OLn+ga7MhMdUxTjHN2yhmXyRhBNenitl=
root@linux10.mlab.internal
```

Figure 16 - Copy the contents of the public certificate

The public certificate now needs to be imported into the Data Domain system. Log on to the Data Domain appliance with an account that has admin privileges. For this example, a new local Data Domain user called *fastcopy* was created and used. Once logged on to the Data Domain as the user *fastcopy*, run the `adminaccess add ssh-keys` command and paste the public key contents as shown in Figure 17.

```
fastcopy@dd01# adminaccess add ssh-keys
Enter the key and then press Control-D, or press Control-C to cancel.
ecdsa-sha2-nistp256 AAvBE2VjZHNhLXNoYTItbmlzdHAyNTYrtAAIbmlzdHAyNTYAAABBBH2/QOSk5y3b/ntSD8bki-
XcW/JM0a0+WCMF9EcYJYvbnlvsiaXpnRZ8NHU1OLn+ga7MhMdUxTjHN2yhmxYrhBNenitl=
root@linux10.mlab.internal
SSH key accepted.
fastcopy@dd01#
fastcopy@dd01# adminaccess show ssh-keys

User "fastcopy" :
 1 ecdsa-sha2-nistp256
AAvBE2VjZHNhLXNoYTItbmlzdHAyNTYrtAAIbmlzdHAyNTYAAABBBH2/QOSk5y3b/ntSD8bki-
XcW/JM0a0+WCMF9EcYJYvbnlvsiaXpnRZ8NHU1OLn+ga7MhMdUxTjHN2yhmxYrhBNenitl=
root@linux10.mlab.internal
```

Figure 17 - Storing the public key into the Data Domain appliance

The final step is to verify if the certificate has been successfully installed and this is done from the Linux server as shown in Figure 18. The verification will be successful if the SSH connection is established without prompting for a password and a basic command can be pass-processed.

```
[root@linux10 .ssh]# ssh fastcopy@dd01.mlab.internal
Data Domain OS
Last login: Tue Jan 7 15:24:12 AEDT 2020 from 192.168.100.89 on pts/2

Welcome to Data Domain OS 6.2.0.35-635767
-----
fastcopy@dd01# quit
Connection to dd01.mlab.internal closed.
[root@linux10 .ssh]#
[root@linux10 .ssh]# ssh fastcopy@dd01.mlab.internal system show uptime
Data Domain OS
17:07:25 up 9 days, 3:12, 3 users, load average: 3.10, 3.17, 3.21
Filesystem has been up 9 days, 03:01.
```

Figure 18 - Verify that the certificate has been successfully imported to the Data Domain

The Linux server is now ready to run scripts in a non-interactive manner.

Sample Shell Script to Create and Remove Fast Copy Instances

The Shell script in Figure 19 was created to provide an example of how to automate creation of Fast Copies of the application MTree and the deletion of them after the retention period has expired. This is by no means a full featured script, however, it enables this solution to be tested without the need to create a script from scratch. Its main focus is to validate that this solution is viable for the organizations' environment for creating immutable data protection copies. A copy of the script can be downloaded from <https://gitlab.com/mvandersteen/fastcopy-rl>.

```
# This Shell script is designed to run from a Linux host and has been tested on a CentOS 7 server
```

```

# The following 7 variables need to be set for this script to work
DATADOMAIN='fqdn'
# FQDN or IP address of Data Domain; DATADOMAIN='datadomain.domain.local'
DDACCOUNT='account'
# The Data Domain account used to create the SSH certificate configuration; eg 'fastcopy'
APPMTREE='/data/col1/....'
# Data Domain MTree location used by the Application; APPMTREE='/data/col1/test'
RLMTREE='/data/col1/..../'
# Data Domain MTree location of the 'Retention Lock' MTree; RLMTREE='/data/col1/test-rl/'
NFSMOUNT='/mnt/nfs/....'
# NFS export on Linux host of 'Retention Lock' MTree on the Data Domain; NFSMOUNT='/mnt/nfs/test-rl'
OUTPUT='.../..../'
# Directory location of the output files; OUTPUT='/tmp/script_output/test/'
RLDAYS=28
# Retention period in days as defined in the Data Domain 'automatic retention period' parameter for the
'Retention Lock' MTree; RLDAYS=28
# Note that a backslash is required at the end of the RLMTREE and OUTPUT variables

# Grabbing the current date/time and assigning it to the variable DATE
DATE=$(date +%Y%m%d-%H:%M)

# Echo the DATE stamp to the script-log text file
echo "Date and time Format to be used: $DATE" > "$OUTPUT$DATE"-script-log.txt

# Echo the Fast Copy command of the application MTree (APPMTREE) to the 'Retention Lock' MTree (RLMTREE)
in the script-log text file
echo "Fast Copy of $APPMTREE to $RLMTREE$DATE" >> "$OUTPUT$DATE"-script-log.txt

# Create a Fast Copy of the application MTree (APPMTREE) and save it to the 'Retention Lock' MTree (RLMTREE)
via an SSH connection
# Output from this command is captured to the script-log text file
ssh $DDACCOUNT@$DATADOMAIN fileys fastcopy source $APPMTREE destination $RLMTREE$DATE >>
"$OUTPUT$DATE"-script-log.txt

# Echo Fast Copy directories that were created more than the number of days as specified in variable (RLDAYS)
for log purposes
echo "Finding directories in $RLMTREE created more than $RLDAYS days ago" >> "$OUTPUT$DATE"-script-
log.txt

# Find Fast Copy directories that were created more than the number of days as specified in variable (RLDAYS)
find $NFSMOUNT -maxdepth 1 -type d -ctime +$RLDAYS ! -path '*snapshot*' -printf '%p\n' >> $OUTPUT$DATE-
directories.txt

# Check to see if the directories file contains any directory listings
if [ -s $OUTPUT$DATE-directories.txt ]
then
    echo "$OUTPUT$DATE-directories.txt contains directory listing" >> "$OUTPUT$DATE"-script-
log.txt

    # Delete the directories found and listed in the $OUTPUT$DATE-directories.txt files
    # Every directory found is read and then deleted
    cat $OUTPUT$DATE-directories.txt | while read LINE
    do
        echo "Found directory: " $LINE >> "$OUTPUT$DATE"-script-log.txt
    done

```

```

rm -rf $LINE
# running $? command to get the output of the above listed rm command.
# If a 0 is returned, the directory was successful deleted
# if a 1 is returned, then the deletion command was not successfully
if [ $? -eq 0 ]
    then
        echo "Successfully deleted directory: " $LINE >> "$OUTPUT$DATE"-
script-log.txt
    else
        echo "Failed to deleted directory: " $LINE >> "$OUTPUT$DATE"-
script-log.txt
    fi
done
else
    echo "No directories found" >> "$OUTPUT$DATE"-script-log.txt
fi
echo "End of script: " $DATE >> "$OUTPUT$DATE"-script-log.txt
# end of script

```

Figure 19 - Sample Shell script to automate the creation and deletion of Fast Copy instances

Scheduling the Script to Run Automatically

There are numerous utilities available designed to run scripts or perform tasks based on a defined date and time schedule. If for testing purposes, a scheduling utility is not available then setting up a Cron Job on a Linux server is an easy and convenient option. An overview of the required format is provided in Figure 20 as detailed in the Wikipedia Cron page¹¹.

```

# |----- minute (0 - 59)
# | |----- hour (0 - 23)
# | | |----- day of the month (1 - 31)
# | | | |----- month (1 - 12)
# | | | | |----- day of the week (0 - 6) (Sunday to Saturday;
# | | | | | 7 is also Sunday on some systems)
# | | | | |
# | | | | |
# * * * * * command to execute

```

Figure 20 - Format and Parameters required for the Cron Job

To configure a Cron job, run the `crontab -e` command on the Linux server as shown in Figure 21 and a text editor is presented where the parameters for the Cron job are entered.

```

[root@linux10 /]# crontab -e
crontab: installing new crontab
[root@linux10 /]#
[root@linux10 /]# crontab -l
0 */8 * * * /tmp/script/fastcopy/fastcopy_test-v1.sh

```

Figure 21 - Configuring a Cron Job to schedule the Fast Copy Script

The example crontab command indicates that the `fastcopy_test-v1.sh` script is run from listed location at 0 minutes, every 8 hours, every day. While the frequency of the script for creating immutable data copies is dependent on an organizations requirements, the minimum recommended frequency would be once per day.

Immutable Data Protection Example for Avamar

Dell Technologies customers looking to provide immutable copies of their backup data to meet a Better level of protection can do so with NetWorker and PowerProtect Data Manager. However, those using Avamar don't have this option and at the time of writing this article, there is no integrated ability to secure the data protected by Avamar with Data Domain Retention Lock.

The way Avamar writes and updates data stored on the Data Domain MTree is the reason why it cannot natively leverage Retention Lock. The likelihood is there would other applications that would fall into this situation. Hence, sharing this solution would provide organizations the ability to store immutable data copies without directly impacting the application itself.

In this section, we look at an example of protecting Avamar data, the requirements and how to recover in the unlikely event that data has been compromised as a result of a cyber event or by the actions of bad actors.

It is recommended to work with a Dell Technologies Data Protection Systems Engineer if considering this solution and that any recovery process of Avamar requires assistance from the Dell Technologies Support Teams.

Solution Requirements

There are several requirements and considerations for successful rollback of Avamar when implementing this solution.

Avamar Checkpoints

A mandatory requirement for this solution is the ability for Avamar to save its checkpoints to the Data Domain MTree used to store the backup data. The checkpoint is performed at the start of the Avamar maintenance window, typically around 9am daily. A snapshot is created when the metadata stored on Avamar and backup data stored on Data Domain are in a consistent state. This checkpoint can be saved to the Data Domain appliance when a single Avamar node or the virtual edition is deployed. The checkpoint stored on Data Domain is validated and when successfully completed, provides a recovery point for Avamar.

Having a valid Avamar checkpoint on Data Domain is the key requirement for a successful rollback operation to allow access to immutable data copies. For day-to-day recovery operations of client data, a rollback is not required. However, it provides a complete disaster recovery of Avamar and the data from a 'Retention Lock' Fast Copy instance.

Unfortunately, this proposed solution is not compatible with an Avamar grid (cluster of 3 or more Avamar nodes) and does not have the option to save the checkpoint to the Data Domain appliance. With an Avamar grid, the checkpoint is stored and written across all nodes that form the cluster.

Metadata stored on Data Stripes

A Dell Technologies white paper titled *Av-DD System: Store Metadata on Data Stripe*¹² provides a detailed explanation of a feature that allows for metadata to be stored on a data stripe. Leveraging this feature enables capacity provided to the single Avamar node or Virtual Edition to be used 100% for metadata.

The command to check the status of this option and to enable it is shown in Figure 22. For detailed information on this feature, review the referenced whitepaper.

```
admin@ave03:~/>: avmaint config --ava |grep checkdiratomicrefs
  checkdiratomicrefs="false"
admin@ave03:~/>:
admin@ave03:~/>: avmaint config --ava checkdiratomicrefs="true"
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<gsanconfig checkdiratomicrefs="true"/>
admin@ave03:~/>:
admin@ave03:~/>: avmaint config --ava |grep checkdiratomicrefs
  checkdiratomicrefs="true"
```

Figure 22 - Command to enable Avamar Metadata to be Written to Data Segments

Avamar Recovery Instance

It is recommended to use an Avamar recovery server, instead of the using the original Avamar for recovery operation. With a recovery instance of Avamar, it can be used to rollback to Fast Copy instances from the 'Retention Lock' MTree, without impacting the original Avamar server.

In the rollback process a new Virtual Edition of Avamar (Avamar recovery instance) is deployed with storage which matches the size of the original Avamar server. With this newly deployed Avamar server, a full disaster recovery is performed by rolling back to a validated checkpoint instance.

This process is outlined in the Support Technical Document titled *Restoring Dell EMC Avamar Checkpoint Backups from a Dell EMC Data Domain System After a Single Node Avamar Failure*¹³ and recommended to use Dell Technologies Support or Professional Services. This document and the commands it references will be used during the example Avamar recovery scenario.

Environment Overview

To help identify the elements used in this example, the following components have been deployed:

- *ave03.mlab.internal* – Production Avamar Virtual Edition server running Avamar 19.2
- *mg01.mlab.internal* – Windows 2012 R2 server protected by Avamar using a standard file agent
- *dd01.mlab.internal* – Data Domain appliance running Operating System version 6.2.0.35
- *linux10.mlab.internal* – CentOS 7 Linux server with NFS mount to the 'Retention Lock' MTree

An Avamar recovery server with the same name and IP address of the production Avamar Virtual Server has been deployed to the VMware environment called *ave03-recovery* as shown in **Figure 23**. This VM cannot be running while the production Avamar server is in use. It is only deployed to enable fast recovery if the need arises.

It is vital that the storage presented to this recovery Avamar server is the same as the production Avamar instance.

Name ↑	State	Status	Provisioned Space	Used Space	Host CPU	Host Mem
AVE03	Powered On	✓ Normal	917.08 GB	24.2 GB	438 MHz	5.84 GB
AVE03-Recovery	Powered Off	✓ Normal	917.69 GB	11.12 GB	0 Hz	0 B

Figure 23 - Avamar Servers as deployed in VMware

Protecting Avamar Backups

Before the Fast Copy instances are made of the Avamar server and associated backup data stored on Data Domain, system properties should be collected.

Avamar System Properties

Using an SSH connection to the Avamar server, run the command `mccli server show-prop` as shown in **Figure 24**. In reviewing the command output, the System ID and the completion of the last valid checkpoint are provided. The System ID as highlighted in yellow, is the Unix Epoch time when Avamar was configured and it is also used in the Data Domain MTree name. Use this System ID to identify the Data Domain MTree used by Avamar.

```
admin@ave03:~/>: mccli server show-prop
0,23000,CLI command completed successfully.
```

Attribute	Value
State	Full Access
Active sessions	0
Total capacity	562.4 GB
Capacity used	843.6 MB
Server utilization	0.1%
Bytes protected (client pre-comp size)	16.2 GB
Bytes protected quota (client pre-comp size)	Not configured
License expiration	2020-04-01 13:19:13 AEDT
Time since Server initialization	23 days 01h:35m
Last checkpoint	2020-01-25 09:03:24 AEDT
Last validated checkpoint	2020-01-25 09:00:45 AEDT
System Name	AVE03.MLAB.INTERNAL
System ID	1577931553@00:50:56:8B:F7:0B
HFSAddr	ave03
HFSPort	27000
IP address	x.x.x.x
Number of nodes	1
Nodes Online	1
Nodes Offline	0
Nodes Read-only	0
Nodes Timed-out	0

Figure 24 - System properties of Avamar server

Confirming Checkpoint Validation

The location of the checkpoint must be confirmed to ensure that it is being written to the Data Domain. Recall that this is only possible for single Avamar Node or the Virtual Edition instance. Log on to the Avamar HTML 5 Web UI and navigate to Administration | system | Server Management | checkpoints. Review the information to confirm that the checkpoints are indeed being written to the Data Domain as shown in Figure 25.

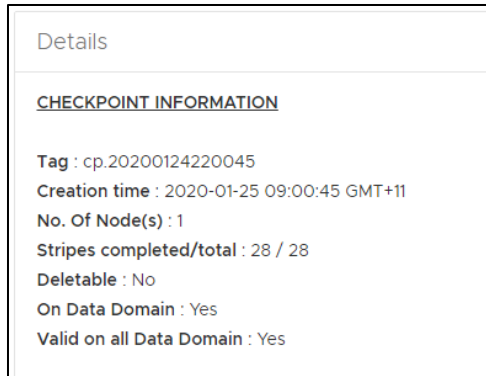


Figure 25 - Avamar checkpoint details

Create and Configure the ‘Retention Lock’ MTree

Create an MTree on the Data Domain with the same name as the Avamar MTree with the suffix of *-rl*; this will be known as the ‘Retention Lock’ MTree. The Avamar MTree will include the System ID as displayed in Figure 24.

Once created, enable automatic Retention Lock on the newly created MTree with the desired settings. For this environment, a retention period of 7 days was configured in Governance mode as shown in Figure 26. For physical Data Domain appliances, either Governance or Compliance mode can be used, while Virtual Editions of Data Domain only support Governance mode.

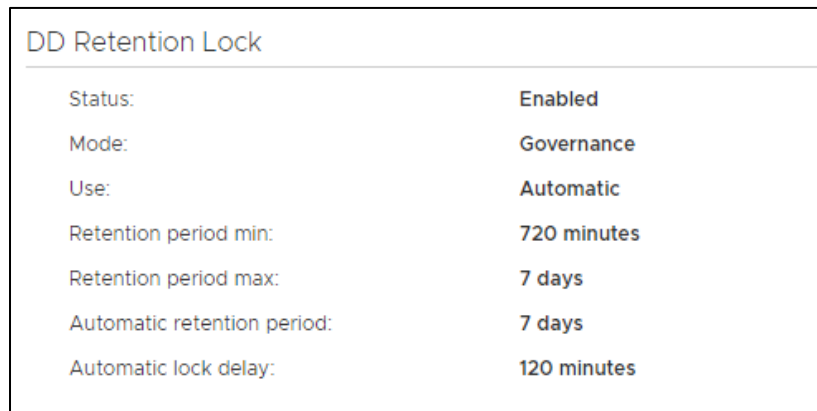


Figure 26 - Retention Lock parameters

Scheduling the Fast Copy Operations

The timing of the Fast Copy operations will depend on when the checkpoint validation is completed. In this environment the checkpoint is validated a few minutes past 9am daily. The time taken to validate the checkpoint will be different for every environment. Note that it is important to ensure that the Fast Copy operation is performed after the checkpoint validation is completed. For this lab environment, the Fast Copy operation was scheduled to start at 10 minutes past 10am every day.

Confirmation of Fast Copy Operations

A simple method to confirm that the Fast Copy operations are successful is to mount an NFS export of the ‘Retention Lock’ MTree to a Linux Host. This also provides a method to identify valid checkpoints and

remove expired Fast Copy instances via a script. Using a WinSCP, there are Fast Copy instances available from 18th January 2020 through to 25th January 2020 as shown in Figure 27.

With the environment running for a few weeks and daily Fast Copy instances being created, a list of directories should be visible via the mounted NFS export as shown in the figure below using WinSCP.

Name	Size	Changed
20200118-10:10		11/01/2020 8:52:14 AM
20200119-10:10		11/01/2020 8:52:14 AM
20200120-10:10		11/01/2020 8:52:14 AM
20200121-10:10		11/01/2020 8:52:14 AM
20200122-10:10		11/01/2020 8:52:14 AM
20200123-10:10		11/01/2020 8:52:14 AM
20200124-10:10		11/01/2020 8:52:14 AM
20200125-10:10		11/01/2020 8:52:14 AM

Figure 27 - Fast Copy instance directories

Available Backups in Lab Environment

Before stepping through the Avamar recovery process, the recovery options available for *mg01.mlab.internal* are shown in Figure 28. Remember that the checkpoint validation is performed during the maintenance windows following the nightly backups. This means that a checkpoint created at 10:10am on the 25th January 2020, provides restore capability for data protected during the night of the 24th January 2020.

	Location	Number	Date&Time	Plugin	Size	Server	Type
<input checked="" type="radio"/>	LOCAL	24	2020-01-24 22:06:23 GMT+11	Windows File System	16.23 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	23	2020-01-23 22:06:34 GMT+11	Windows File System	16.24 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	22	2020-01-22 22:06:28 GMT+11	Windows File System	16.24 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	21	2020-01-21 22:06:18 GMT+11	Windows File System	16.24 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	20	2020-01-20 22:06:28 GMT+11	Windows File System	16.23 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	19	2020-01-19 22:06:43 GMT+11	Windows File System	16.23 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	18	2020-01-18 22:06:17 GMT+11	Windows File System	16.24 GB	DD - dd01.mlab.internal	Full

Figure 28 - MG01 recovery options

Recovering Avamar and Protected Data

The recovery process shown in this section is based on a scenario where the Avamar server (*ave03*) and several production servers, including *mg01* were compromised around midday on the 24th January 2020 and not discovered until the morning of the 25th. An executive decision has been made to use the recovery Avamar server and rollback to the checkpoint taken in the morning of the 24th. Once completed, data stored on *mg01* server can be recovered.

A high-level overview of the recovery process is;

- Shutdown the compromised Avamar server
- Confirm the ID of a valid checkpoint taken on the morning of the 24th
- Fast copy to the Avamar MTree the identified Fast Copy instance of morning of the 24th
- Perform the Avamar server recovery from the identified checkpoint
- Recovery data to *mg01* server

Shutdown the Avamar Server

Connect to the Avamar server via SSH, logging on as admin and run the command *dpnctl stop* to gracefully stop all Avamar services as shown in Figure 29. Once all services have been stopped, *su root* and *shutdown* the Avamar server.

```
admin@ave03:~>: dpnctl stop
Identity added: /home/admin/.ssh/admin_key (/home/admin/.ssh/admin_key)
-----
Do you wish to shut down the local instance of EM Tomcat?

Answering y(es) will shut down the local instance of EM Tomcat
      n(o) will leave up the local instance of EM Tomcat
      q(uit) exits without shutting down

y(es), n(o), q(uit/exit): y
dpnctl: INFO: Suspending backup scheduler...
dpnctl: INFO: Backup scheduler suspended.
dpnctl: INFO: Checking for active checkpoint maintenance...
dpnctl: INFO: Terminating hfs integrity maintenance (hfscheck)...
dpnctl: INFO: To monitor progress, run in another window: tail -f /tmp/dpnctl-subsystem-control-action-output-3774
dpnctl: INFO: EM Tomcat shut down.
dpnctl: INFO: Shutting down MCS...
dpnctl: INFO: MCS shut down.
dpnctl: INFO: To monitor progress, run in another window: tail -f /tmp/dpnctl-subsystem-control-action-output-3774
dpnctl: INFO: Shutting down gsan...
dpnctl: INFO: gsan shut down.
admin@ave03:~>:
admin@ave03:~>: su root
Password: *****
root@ave03:/home/admin/#: shutdown
Shutdown scheduled for Sat 2020-01-25 16:34:42 AEDT, use 'shutdown -c' to cancel.
```

Figure 29 - Gracefully shutdown Avamar server

Confirm the ID of a Valid Checkpoint

With a WinSCP connection to *Linux10.mlab.internal* server, the contents of the Fast Copy instance created during the morning of the 24th will be examined. At the root of the 'Retention Lock' MTree, all Fast Copy instances are listed as shown in Figure 30 using the naming convention as discussed in step 3 of the Creating Immutable Data Copies section on page 9.

/mnt/nfs/ave03-rl		
Name	Size	Changed
↑		24/01/2020 5:15:47 PM
20200118-10:10		11/01/2020 8:52:14 AM
20200119-10:10		11/01/2020 8:52:14 AM
20200120-10:10		11/01/2020 8:52:14 AM
20200121-10:10		11/01/2020 8:52:14 AM
20200122-10:10		11/01/2020 8:52:14 AM
20200123-10:10		11/01/2020 8:52:14 AM
20200124-10:10		11/01/2020 8:52:14 AM
20200125-10:10		11/01/2020 8:52:14 AM

Figure 30 - Listing of Fast Copy instances

Navigating into the Fact Copy instance *20200124-10:10* created on the morning of the 24th, several directories are listed specific to Avamar as shown in Figure 31. A brief overview of the relevant directories are as follows;

- cur – contains the backup sets for each client
- GSAN – this contains metadata and associated data to rebuild Avamar and only exists if the option to save the checkpoint to Data Domain is enabled
- STAGING – location where in progress backups are stored and once successfully completed are moved to the cur directory
- VALIDATED – the directory that contains the ID of a valid checkpoint

/mnt/nfs/ave03-rl/20200124-10:10		
Name	Size	Changed
↑		25/01/2020 10:58:23 AM
cur		9/01/2020 8:52:03 AM
DELETED		11/01/2020 8:52:14 AM
GSAN		24/01/2020 9:55:16 AM
STAGING		2/01/2020 5:11:01 PM
VALIDATED		24/01/2020 9:56:25 AM
ddrid	1 KB	24/01/2020 10:58:14 AM

Figure 31 - Avamar specific directories created in the Data Domain MTree

Within the VALIDATED directory two checkpoints IDs are listed as shown in **Figure 32** and with a name in the format of *yyyymmddhhmmss*. Note that the time format of the checkpoint is based on UTC.

/mnt/nfs/ave03-rl/20200124-10:10/VALIDATED		
Name	Size	Changed
↑		11/01/2020 8:52:14 AM
cp.20200123220042		24/01/2020 9:55:45 AM
cp.20200123220322		24/01/2020 9:56:21 AM

Figure 32 - List of available checkpoints

To confirm which checkpoint ID should be referenced for the recovery, navigate into each directory and one of them should contain a file called *checkpointvalid* as shown in **Figure 33**.


/mnt/nfs/ave03-rl/20200124-10:10/VALIDATED/cp.20200123220042		
Name	Size	Changed
		24/01/2020 9:56:25 AM
<input type="checkbox"/> checkpointvalid	1 KB	24/01/2020 10:58:15 AM

Figure 33 - Valid checkpoint

For the purposes of this recovery, the checkpoint ID cp.20200123220042 will be used. This recovery example is using an Avamar server which is configured with a time zone of UTC +11. Adding 11 hours to the checkpoint ID indicates it was taken at 42 seconds past 9am on the 24th of January 2020. This may be a little confusing at first but once the offset to UTC is known, the local time that the checkpoint was taken can be easily calculated.

Fast Copy

Connecting to the Data Domain via an SSH connection, the contents of the identified Fast Copy instance needs to be copied to the Avamar MTree. The Fast Copy command as shown in **Figure 34** copies the contents of /data/col1/avamar-1577931553-rl/20200124-10:10 to /data/col1/avamar-1577931553.

```

sysadmin@dd01# filesys fastcopy source /data/col1/avamar-1577931553-rl/20200124-10:10 destination
/data/col1/avamar-1577931553

Destination "/data/col1/avamar-1577931553" already exists.
Proceeding will overwrite its content with "/data/col1/avamar-1577931553-rl/20200124-10:10".
Are you sure? (yes|no) [no]: yes

ok, proceeding.

Creating snapshot "FASTCOPY-2020-01-25-17-39-23" with one-hour retention period...done
Use this snapshot to recover in case of a mistake.

(00:00) Waiting for fastcopy to complete...
Fastcopy status: fastcopy /data/col1/avamar-1577931553-rl/20200124-10:10 to /data/col1/avamar-
1577931553: deleted 145 files, 17 directories; copied 131 files, 17 directories in 1.22 seconds

```

Figure 34 - Fast Copy command to commence the recovery process

The time required to copy the contents to the MTree will depend on the volume of data protected by Avamar.

Powering on the Recovery Avamar Server

With the Avamar server shut down and the Fast Copy instance copies to the Avamar MTree, the recovery Avamar server can now be powered on. Log on to the Recovery Avamar server via SSH and confirm that all services have started by running the `dpnctl status` command as shown in **Figure 35**.

```

admin@ave03:~/>: dpnctl status
Identity added: /home/admin/.ssh/admin_key (/home/admin/.ssh/admin_key)
dpnctl: INFO: gsan status: up
dpnctl: INFO: MCS status: up.
dpnctl: INFO: emt status: up.
dpnctl: INFO: Backup scheduler status: up.
dpnctl: INFO: Maintenance windows scheduler status: suspended.
dpnctl: INFO: Unattended startup status: enabled.

```

```
dpnctl: INFO: avinstaller status: up.
dpnctl: INFO: ConnectEMC status: up.
dpnctl: INFO: ddrmaint-service status: up.
```

Figure 35 - Confirming the state of Avamar services

For the purposes of this recovery, the command `mccli server show-prop` as shown in **Figure 36** confirms that the recovery server has the same total capacity and name. Note that the recovery Avamar server ID of 1579932265 is not the same as the original Avamar server of 1577931553. During the Avamar recovery process, the recovery Avamar server will inherit the system ID of 1577931553 from the checkpoint data contained in the GSAN directory.

```
admin@ave03:~/>: mccli server show-prop
0,23000,CLI command completed successfully.
Attribute                                     Value
-----
State                                         Full Access
Active sessions                               0
Total capacity                               562.4 GB
Capacity used                                 0 bytes
Server utilization                             0.0%
Bytes protected (client pre-comp size)        0 bytes
Bytes protected quota (client pre-comp size)  Not configured
License expiration                            2020-04-24 16:04:25 AEST
Time since Server initialization               0 days 00h:25m
Last checkpoint                               2020-01-25 17:19:55 AEDT
Last validated checkpoint                     No validated checkpoints.
System Name                                   AVE03.MLAB.INTERNAL
System ID                                     1579932265@00:50:56:8B:DD:99
HFSAddr                                       ave03
HFSPort                                       27000
IP address                                    x.x.x.x
Number of nodes                               1
Nodes Online                                 1
Nodes Offline                                 0
Nodes Read-only                               0
Nodes Timed-out                              0
```

Figure 36 – Avamar Recovery server system properties

Recovering the Avamar Server from the Checkpoint

The procedure detailed in the *Restoring Dell EMC Avamar Checkpoint Backups from a Dell EMC Data Domain System after a Single Node Avamar Failure*¹³ technical note document is used. It is recommended that Dell Technologies Professional Services or Support are engaged to perform the recovery process. Normally this recovery process uses the last validated checkpoint, however, for this Immutable data protection solution a validated checkpoint older than the latest may be used.

This type of recovery should not be required when performing regular recoveries. It is intended to provide data recovery when the Avamar server or its data has been compromised.

The commands used to recover the Avamar server looks for a snapshot of the Avamar MTree with the name of the checkpoint ID. Depending on recovery checkpoint used, the snapshot may not exist resulting in the recovery command not completing successfully. In this case, a snapshot would need to be created

as shown in **Figure 37**. Situations like this is one reason why Dell Technologies Support needs to be engaged.

```
sysadmin@dd01# snapshot create cp.20200123220042 mtree /data/col1/avamar-1577931553
Snapshot "cp.20200123220042" created for mtree "/data/col1/avamar-1577931553".
```

Figure 37 - Creation of checkpoint snapshot on Avamar MTree

The first step in the recovery process requires the Avamar MTree and associated checkpoint to be located on the Data Domain as shown in Figure 38. This command and others are all run as root and performed via an SSH connection on the recovery Avamar server.

```
admin@ave03:~/>: su - root
Password: *****
root@ave03:~/#:
root@ave03:~/#: ddrmaint cp-backup-list --full --ddr-server=dd01.mlab.internal --ddr-user=ddbboost --ddr-
password=*****

===== Checkpoint =====
Avamar Server Name       : ave03
Avamar Server MTree/LSU  : avamar-1577931553
Data Domain System Name  : dd01.mlab.internal
Avamar Client Path       : /MC_SYSTEM/avamar-1577931553
Avamar Client ID         : e6c0 88673496f2f5aed03c45676d9c01c2c062a5
Checkpoint Name          : cp.20200123220042
Checkpoint Backup Date   : 2020-01-24 09:02:18
Data Partitions          : 3
Attached Data Domain systems : dd01.mlab.internal
```

Figure 38 - Identifying the Data Domain MTree and checkpoint

The output in Figure 38 confirms that the correct Data Domain MTree (Avamar Client Path) and the associated Avamar Checkpoint have been discovered. The Avamar rollback process can now begin.

The command *cprestore* is shown in Figure 39 and results in the contents of the GSAN directory associated with the checkpoint being copied to the recovery Avamar server. This process may take some time and is dependent on the number of stripes to be copied.

```
root@ave03:/home/admin/#: cprestore --hfscreatetime=1577931553 --ddr-server=dd01.mlab.internal --ddr-
user=ddbboost --cptag=cp.20200123220042
*****
Version: 1.11.1
Current working directory: /space/avamar/var
Log file: cprestore-cp.20200123220042.log
Checking node type.
Node type: single-node server
Create DD NFS Export: data/col1/avamar-1577931553/GSAN
ssh ddbboost@dd01.mlab.internal nfs add /data/col1/avamar-1577931553/GSAN x.x.x.x
"(ro,no_root_squash,no_all_squash,secure)"
Execute: ssh ddbboost@dd01.mlab.internal nfs add /data/col1/avamar-1577931553/GSAN x.x.x.x
"(ro,no_root_squash,no_all_squash,secure)"
Warning: Permanently added 'dd01.mlab.internal,x.x.x.x' (RSA) to the list of known hosts.
Data Domain OS
Password: *****
Execute output:
```

```

NFS export for "/data/col1/avamar-1577931553/GSAN" added.

Mount NFS path 'dd01.mlab.internal:/data/col1/avamar-1577931553/GSAN' to 'ddnfs_gsan'
Local mount path 'ddnfs_gsan' does not exists... creating it.
Execute: sudo mount -t nfs dd01.mlab.internal:/data/col1/avamar-1577931553/GSAN "ddnfs_gsan" -o
ro,nolock
User provided the checkpoint 'cp.20200123220042'.
Data disks on server: data01 data02 data03
Data disks on local: data01 data02 data03
Directory /data01 exists
Directory /data02 exists
Directory /data03 exists
Verify no checkpoint exists locally.
.....output omitted.....
Restore data02 finished.
.....output omitted.....
Restore data03 finished.
.....output omitted.....
Restore data01 finished.
.....output omitted.....
PID 30506 returned with exit code 0
Finished restoring files in 00:00:20.
Restoring ddr_info.
Copy: 'ddnfs_gsan/cp.20200123220042/DDR_info' -> '/usr/local/avamar/var/DDR_info'
Unmount NFS path 'ddnfs_gsan' in 3 seconds
Execute: sudo umount "ddnfs_gsan"
Remove DD NFS Export: data/col1/avamar-1577931553/GSAN
ssh ddboost@dd01.mlab.internal nfs del /data/col1/avamar-1577931553/GSAN x.x.x.x
Execute: ssh ddboost@dd01.mlab.internal nfs del /data/col1/avamar-1577931553/GSAN x.x.x.x
Data Domain OS
Password: *****
kthxbye

```

Figure 39 - Commence the checkpoint restore process

The next step in the process is to stop the Avamar services using the `dpnctl stop` command as shown in Figure 40.

```

root@ave03:~/#: dpnctl stop
Identity added: /home/admin/.ssh/admin_key (/home/admin/.ssh/admin_key)

-----
Do you wish to shut down the local instance of EM Tomcat?

Answering y(es) will shut down the local instance of EM Tomcat
n(o) will leave up the local instance of EM Tomcat
q(uit) exits without shutting down

y(es), n(o), q(uit/exit): y
dpnctl: INFO: Suspending backup scheduler...
dpnctl: INFO: Backup scheduler suspended.
dpnctl: INFO: Checking for active checkpoint maintenance...
dpnctl: INFO: Terminating hfs integrity maintenance (hfscheck)...

```

```

dpnctl: INFO: To monitor progress, run in another window: tail -f /tmp/dpnctl-subsystem-control-action-output-31519
dpnctl: INFO: EM Tomcat shut down.
dpnctl: INFO: Shutting down MCS...
dpnctl: INFO: MCS shut down.
dpnctl: INFO: To monitor progress, run in another window: tail -f /tmp/dpnctl-subsystem-control-action-output-31519
dpnctl: INFO: Shutting down gsan...
dpnctl: INFO: gsan shut down.

```

Figure 40 - Stopping the Avamar services

The final step in the recovery process is to rollback Avamar to the checkpoint identified in Figure 33 and restart the services. In Figure 41 the `dpnctl start --force_rollback` command is run and answer *yes* if Dell Technologies Support is engaged. Option 3 is selected, enabling the identified checkpoint to be used for the rollback of Avamar. In relation to restoring the local EMS data, answer *no* unless advised otherwise by Support.

```

root@ave03:~/>: dpnctl start --force_rollback
Identity added: /home/admin/.ssh/admin_key (/home/admin/.ssh/admin_key)
-----
Action: starting all
Have you contacted Avamar Technical Support to ensure that this is the right thing to do?

Answering y(es) proceeds with starting all;
n(o) or q(uit) exits

y(es), n(o), q(uit/exit): y
dpnctl: INFO: Checking that gsan was shut down cleanly...
-----
Here is the most recent available checkpoint:
Thu Jan 23 22:00:42 2020 UTC Not Validated

A rollback was requested.
The gsan was shut down cleanly.

The choices are as follows:
 1 roll back to the most recent checkpoint, whether or not validated
 2 (not applicable: no validated checkpoints are available)
 3 select a specific checkpoint to which to roll back
 4 do not restart
q quit/exit

(Entering an empty (blank) line twice quits/exits.)
> 3
-----
Here is the list of available checkpoints:

 2 Sat Jan 25 06:19:55 2020 UTC Not Validated
 1 Thu Jan 23 22:00:42 2020 UTC Not Validated

Please select the number of a checkpoint to which to roll back.
Alternatively:

```

```
q return to previous menu without selecting a checkpoint

(Entering an empty (blank) line twice quits/exits.)
> 1
-----
You have selected this checkpoint:

name: cp.20200123220042
date: Thu Jan 23 22:00:42 2020 UTC
validated: no
age: 1 day, 9 hours

Roll back to this checkpoint?

Answering y(es) accepts this checkpoint and initiates rollback
n(o) rejects this checkpoint and returns to the main menu
q(uit) exits

y(es), n(o), q(uit/exit): y
dpnctl: INFO: Initiating rollback to "cp.20200123220042" with gsan restart (this may take some time)...
dpnctl: INFO: rolling back to checkpoint "cp.20200123220042" and restarting the gsan succeeded.
dpnctl: INFO: gsan started.
dpnctl: INFO: To monitor progress, run in another window: tail -f /tmp/dpnctl-subsystem-control-action-output-2112
dpnctl: INFO: Restoring MCS data...
dpnctl: INFO: MCS data restored.
dpnctl: INFO: Starting MCS...
dpnctl: INFO: To monitor progress, run in another window: tail -f /tmp/dpnctl-mcs-start-output-2112
dpnctl: INFO: MCS started.
-----
Do you wish to do a restore of the local EMS data?

Answering y(es) will restore the local EMS data
n(o) will leave the existing EMS data alone
q(uit) exits with no further action.

Please consult with Avamar Technical Support before answering y(es).

Answer n(o) here unless you have a special need to restore the EMS data, e.g., you are restoring this node from scratch, or you know for a fact that you are having EMS database problems that require restoring the database.

y(es), n(o), q(uit/exit): n
dpnctl: INFO: To monitor progress, run in another window: tail -f /tmp/dpnctl-subsystem-control-action-output-2112
dpnctl: INFO: EM Tomcat started.
dpnctl: INFO: Resuming backup scheduler...
dpnctl: INFO: Backup scheduler resumed.
dpnctl: INFO: No /usr/local/avamar/var/dpn_service_status exist.
dpnctl: INFO: AvInstaller is already running.
dpnctl: INFO: [see log file "/usr/local/avamar/var/log/dpnctl.log"]
```

Figure 41 - Rollback the Avamar server to the identified checkpoint

Upon successful Avamar rollback, run the `mccli server show-prop` command as shown in Figure 42 to confirm that Avamar server ID is the same as the original server.

```
admin@ave03:~/>: mccli server show-prop
0,23000,CLI command completed successfully.
Attribute                                     Value
-----
State                                         Full Access
Active sessions                               0
Total capacity                               562.4 GB
Capacity used                                 843.6 MB
Server utilization                            0.1%
Bytes protected (client pre-comp size)       16.2 GB
Bytes protected quota (client pre-comp size) Not configured
License expiration                           2020-04-01 13:19:13 AEDT
Time since Server initialization              23 days 05h:13m
Last checkpoint                              2020-01-25 17:19:55 AEDT
Last validated checkpoint                    No validated checkpoints.
System Name                                  AVE03.MLAB.INTERNAL
System ID                                    1577931553@00:50:56:8B:F7:0B
HFSAddr                                      ave03
HFSPort                                      27000
IP address                                    x.x.x.x
Number of nodes                              1
Nodes Online                                 1
Nodes Offline                                0
Nodes Read-only                              0
Nodes Timed-out                              0
```

Figure 42 - System properties after rollback process of the Avamar server

The data and time of the last checkpoint provided in Figure 42 do not match those of the identified checkpoint used to rollback the Avamar server. Logging on to the Avamar HTML 5 Web UI and navigating to Administration | system | Server Management | checkpoints shows available checkpoints. The identified checkpoint used for the rollback process has indeed been applied as shown in Figure 43.

Tag	Validated	Creation time	Nodes	Stripes	Validation Start Time	Validation End time	Errors
<input checked="" type="radio"/> cp.20200123220042		2020-01-24 09:00:42 GMT+11	1	28	2020-01-24 09:02:18 GMT+11	N/A	0
<input type="radio"/> cp.20200125061955	?	2020-01-25 17:19:55 GMT+11	1	25	N/A	N/A	0

Figure 43 - Checkpoint of Recovery Avamar Server

Recovery of Client

Based on the scenario, the successfully recovered Avamar server can now commence the recovery of client data. Looking at the recovery options for the server mg01.mlab.internal in Figure 44, data can be recovered from the backup completed at 10:06pm on the 23rd January.

	Location	Number	Date&Time	Plugin	Size	Server	Type
<input checked="" type="radio"/>	LOCAL	23	2020-01-23 22:06:34 GMT+11	Windows File System	16.24 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	22	2020-01-22 22:06:28 GMT+11	Windows File System	16.24 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	21	2020-01-21 22:06:18 GMT+11	Windows File System	16.24 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	20	2020-01-20 22:06:28 GMT+11	Windows File System	16.23 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	19	2020-01-19 22:06:43 GMT+11	Windows File System	16.23 GB	DD - dd01.mlab.internal	Full
<input type="radio"/>	LOCAL	18	2020-01-18 22:06:17 GMT+11	Windows File System	16.24 GB	DD - dd01.mlab.internal	Full

Figure 44 - Available Recovery Options for Server mg01.mlab.internal

The final task shown in Figure 45 indicates a successful recovery of data to the server mg01.mlab.internal.

<input type="checkbox"/>	Status	Client	Started	Processed Bytes	Plugin	Type
<input type="checkbox"/>	<input checked="" type="radio"/> Completed	mg01.mlab.internal	2020-01-26 11:19:11 GMT+11	243.09 MB	Windows File System	Restore

Figure 45 – Data Successfully Recovered for Server mg01.mlab.internal

Summary

With the ever-increasing risk posed by Cyber-related attacks and insider threat of bad actors, ensuring data is protected and recoverable is more important than ever. The solution provided in this article leverages the power of Data Domain Retention Lock and its Fast Copy feature to provide organizations a better level of data protection.

With Fast Copy operations, independent full copies of the MTree used by an application are stored in an alternative MTree protected by Data Domain's Automatic Retention Lock feature. This ensures that data is stored in an immutable manner, and it may also include the application's configuration data, providing full Disaster Recovery capability of the application and its data.

References

1. https://i.dell.com/sites/csdocuments/Learn_Docs/es/co/jorge-espinoza-cyber-security.pdf
2. <https://www.dellemc.com/en-us/data-protection/cyber-recovery-solution.htm>
3. <https://www.dellemc.com/en-us/data-protection/powerprotect-dd-series.htm>
4. <https://www.dellemc.com/en-us/data-protection/data-protection-suite/index.htm>
5. <https://www.dellemc.com/en-us/data-protection/powerprotect-software.htm>
6. <https://www.sec.gov/rules/interp/34-47806.htm>
7. <https://linuxize.com/post/linux-touch-command/>
8. https://support.emc.com/docu91804_Data-Domain-Operating-System-6.2-Administration-Guide.pdf?language=en_US
9. https://support.emc.com/docu91805_Data-Domain-Operating-System-6.2-Command-Reference-Guide.pdf?language=en_US
10. <http://thebackupsblog.blogspot.com/2014/09/client-side-scripting-on-datadomain.html>
11. <https://en.wikipedia.org/wiki/Cron>
12. https://support.emc.com/docu91873_AV-DD-System:-Store-Metadata-on-Data-Stripe.pdf?language=en_US
13. https://support.emc.com/docu48257_Restoring-Avamar-Checkpoint-Backups-from-a-Data-Domain-System-After-a-Single-Node-Avamar-Failure-Technical-Note.pdf?language=en_US

Dell Technologies believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” DELL TECHNOLOGIES MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying and distribution of any Dell Technologies software described in this publication requires an applicable software license.

Copyright © 2020 Dell Inc. or its subsidiaries. All Rights Reserved. Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.