# CENTERA TO ECS MIGRATION

## Overview


### ABSTRACT

This white paper provides an overview for customers interested in migrating applications and data from their existing Dell EMC® Centera™ storage to Dell EMC Elastic Cloud Storage (ECS™).


October 2016

# TABLE OF CONTENTS

VERSION HISTORY ..... 5

EXECUTIVE SUMMARY ..... 5

AUDIENCE ..... 5

SCOPE ..... 5

OVERVIEW ..... 5

CENTERA BASICS ..... 5

CENTERA RETENTION AND COMPLIANCE ..... 6

CENTERA CONSTRUCTS ..... 6

CENTERA DATA PROTECTION AND REPLICATION ..... 6

CENTERA CLUSTER ACCESS ..... 6

ECS BASICS ..... 7

ECS CAS ..... 7

ECS SUPPORT FOR CAS COMPLIANCE ..... 8

ECS CAS NAMING ..... 8

ECS CAS INTERNALS ..... 8

CORE DIFFERENCES BETWEEN CENTERA AND ECS ..... 9

MIGRATION REQUIREMENTS ..... 9

ECS SYNC ..... 9

ECS TRANSFORMATION ENGINE ..... 10

TRANSFORMATION ..... 11

TRANSFORMATION WITH MIGRATION ..... 12

TRANSFORMATION AND MIGRATION PHASES ..... 12

Phase 1: Configuration ..... 12

Phase 2: Pre-Check ..... 12

Phase 3: Enumeration ..... 13

Phase 4: Indexing ..... 14

Phase 5: Migration ..... 14

Phase 6: Reconciliation ..... 14

Phase 7: Re-Migration ..... 14

Phase 8: Deleting ..... 14

**CONSIDERATIONS** .......... **16**

**SUMMARY** .......... **17**

**APPENDIX A - CREATING A CAS BUCKET ON ECS** .......... **17**

CREATING A CAS BUCKET ON ECS .......... 17

# VERSION HISTORY

| Version | Part Number | Date | Details |
|---|---|---|---|
| 1 | H15115 | May 2016 | Initial release. |
| 1.0.1 | H15115.1 | August 2016 | Single-instanced objects, 2.2.1 HF1 #10465. Server-side check to avoid copying SIS more than once. Improves related ECS Sync behavior. |
| 1.0.2 | H15115.2 | September 2016 | Additional clarity on single instancing with ECS Sync. |
| 1.0.3 | H15115.3 | October 2016 | Add 3.0 support for ARM. |

Table 1

# EXECUTIVE SUMMARY

Centera was introduced in 2001 as Dell EMC's first object storage platform. It was built to store billions of objects for long periods and primarily intended for unstructured and immutable content. Centera has a track record of being highly stable and reliable. Several thousand customers are using Centera, and many integrated partners and applications are still actively in use today.

Elastic Cloud Storage (ECS) is Dell EMC's latest unstructured storage platform. ECS also provides Content-Addressable Storage (CAS), like Centera, as well as support for all of the major object protocols (S3, Swift and Atmos), and both NFS and HDFS. ECS is built around modern HTTP(S) REST APIs and is grounded in a highly dense, low cost deployment model.

This paper covers a basic overview of each product and the available options to migrate application workload from Centera to ECS.

## AUDIENCE

This white paper is intended for customer and field consumption to assist in providing a high level overview of options for those who are considering transitioning applications from Centera to ECS.

## SCOPE

This paper provides a general overview of the tools available for migrating data from Centera to ECS. It is not intended to provide a step-by-step guide to implement a migration or describe in detail the subtle nuances of each tool. Although the ECS Sync tool can also be used to migrate data from Atmos to ECS, in certain scenarios, Atmos and its future expansion utility, which will allow for ECS to serve as expansion capacity for an existing Atmos Resource Management Group (RMG), are both out of the scope of this paper.

# OVERVIEW

Two options are provided by Dell EMC to assist in migrating data from Centera to ECS:

- ECS Sync, a standalone java application
- The Centera Transformation and Migration feature, introduced natively in the ECS software

This overview provides the basics of both Centera and ECS and provides options and requirements for migrations.

## CENTERA BASICS

Centera is a CAS-only dedicated system. An application delivers a data object to the Centera API which calculates a Content Address (CA) from the object's binary representation, its unique digital fingerprint. This CA and metadata about the object are inserted into an XML file, called a Content Descriptor File (CDF), or C-Clip for short, which in turn has its content address calculated. The C-Clip CA is returned to the application once the CDF and application data, referred to as a BLOB (Binary Large Object), have been successfully stored and protected in the storage. A feature called single instancing storage is available on Centera. Its major benefit is that if the same BLOB is written multiple times, only one instance of it will be stored, and each referencing application gets its own C-Clip CA pointing to the common object.

## CENTERA RETENTION AND COMPLIANCE

Centera has mechanisms in place for the enforcement of data retention and has three compliance levels, Basic, Governance Edition (GE) and Compliance Edition Plus (CE+), which are governed at the cluster level.

In basic mode Centera acts like a standard magnetic storage. An object marked for deletion is deleted immediately. In GE mode, active retention ensures availability of objects for a configurable period of time. An object marked for deletion is not deleted until the retention period passes. In CE+ mode, similar to GE, objects also have a retention period, however, the default retention period is infinite and is applied in the absence of an application defined retention period.

Retention is a major component of Centera storage and retention policies can be applied to a cluster, pool, or object. Retention can be fixed, event-based, or on a timeframe set by a feature named *Litigation Hold*. The period specifies the time the object is immutable on the system. Only with the PRIVILEGED_DELETE capability can an object under a retention period be removed and if so it is done so in an audited manner. The delete function works at the C-Clip level. Once all pointers to a C-Clip have been deleted, the space occupied by the data objects will be reclaimed by a background process known as “garbage collection”. Optionally, for privacy or security reasons, the deleted data may be overwritten (“shredded”) as the space is reclaimed.

## CENTERA CONSTRUCTS

A Centera pool is a Software Development Kit (SDK) object that represents one or more clusters. Virtual pools are created for logical data isolation and authentication is done at the pool level. Reporting and replication are also done at the pool level. An application must open a pool by providing a series of node IP addresses and access profile credentials for the desired set of clusters. The first accessible IP address in the list represents the primary cluster, while subsequent IP addresses are considered the secondary clusters (assuming that they represent distinct clusters). The pool object also auto-discovers any replica clusters that are configured via the primary or secondary clusters. Applications derive their access profile credentials from a Pool Entry Authorization (PEA) file that was generated by a storage administrator after the creation or update of an access profile. Generally an application will write to one pool at a time, keeping existing information in read-only state in other pools. Pools support iteration, access policies, and replication policies.

An application pool is a pool bound to one or more application profiles. The system administrator can create, delete, and update custom application pools. A default pool is built-in and cannot be deleted. During the creation of an application profile it is possible to refer to a custom application pool and assign it as the home pool. All new data from the application profile will then be written to its home pool. If the application profile has no home pool, the data will be written to the default pool. The default pool is also an application pool and contains every C-Clip on the cluster that is not contained in a custom application pool. Although one application can only have one home pool at a time, it is possible that multiple applications share the same home pool. Every C-Clip on the cluster belongs to exactly one application, custom or default pool. Custom application pools are identified by a unique ID which cannot be changed. A custom application pool also has a pool name which must be unique on the cluster.

## CENTERA DATA PROTECTION AND REPLICATION

Centera and the API work together to guarantee end-to-end content authenticity. This is done by computing hashes for all data put on the wire. If a hash check fails, the transaction fails, and in this way Centera never stores or returns corrupted data.

There are two data protection schemes on Centera, mirroring and parity and they are only relevant to migration in the sense of the amount of overhead difference between data on Centera and on an ECS. Replication is done in a uni-directional manner and failover is supported. Centera uses an eventual consistency model for replication.

## CENTERA CLUSTER ACCESS

Centera is only accessible for applications via the Access API, a Dell EMC proprietary protocol named HPP, which uses port 3218. No modern APIs exist for Centera but there is a java-based viewer, Centera Viewer, which is not compatible with ECS. Applications connect to access nodes which in turn access data on dedicated storage nodes that do not directly connect to the customer network. For communication Dell EMC provides an SDK that supports transactions for read, write, query, etc. For management there is a Management API (MAPI) used by Centera Viewer, the CLI and a java console. Health reports and alerts are available over SMTP and some router communication takes place over the Simple Object Access Protocol (SOAP).

A system administrator grants application access at the pool level, where pools represent collections of clips, profiles represent applications. Profiles are a means to enforce authentications and authorization. The system administrator determines which applications have access to a cluster and what operations they can perform. An application can only log into Centera if a profile for that application has been created on the cluster and the credentials for that profile have been made available to the application server.

| CENTERA CAPBILITY | CAPABILITY DESCRIPTION |
|---|---|
| r: read | Read C-Clips |
| w: write | Write C-Clips |
| d: delete | Delete C-Clips |
| q: query | Query C-Clips using a time-based query |
| e: exists | Check for the existence of C-Clips |
| D: privileged delete | Delete C-Clips, overruling retention periods |
| c: clip copy | Copy C-Clips |
| h: retention hold | Set and remove litigation holds on C-Clips |

Table 2

The administrator determines which applications have access to a pool and what operations they can perform on the pool data by assigning a set of capabilities. Capabilities are rights granted by the system administrator to an access profile on a pool. These capabilities determine which operations the application using the access profile can perform on C-Clips belonging to that pool. Table 2 gives an overview of the list of capabilities.

## ECS BASICS

ECS software is built using a layered approach. Each layer is independently scalable, highly available, and ensures no single point of failure. The data services layer in ECS offers application access via all major object protocols, S3, Swift, CAS and Atmos, and NFS and HDFS. Protocol heads within this layer communicate with applications and leverage underlying storage engine API for read/write access to disks. In the storage layer the storage engine, which is the core of the ECS, is responsible for data protection and replication. Regardless of protocol access, all user data, and even all system data, is stored in logical chunks of 128 MB. The 128 MB chunks are written in an append-only fashion so no data is overwritten or modified as in traditional filesystems. With this there is no locking required for I/O and no cache invalidation is required. ECS uses strong consistency and replication is allowed over up to eight all-active geo-locations. By being strongly consistent, ECS avoids the common problem with eventual consistency models such as global namespace with read-only replicas and difficulty of writes by applications. This simplifies development.

## ECS CAS

CAS data on ECS is protected using the common ECS replication mechanism which fundamentally differs from the replication method in Centera. ECS replication uses a low-level object-unaware data copy mechanism. It stores and replicates data at the chunk level across all sites in a replication group. Applications can access replicated data from any site in a replication group, at any time. Centera replicates data at the C-Clip level and can have one optional disaster recovery (DR). In the event of an issue at the primary location, applications must be failed over to the DR site for access to data. Thus, the operational content failover use case is not applicable to ECS CAS as it is with Centera. The connection failover and operational network failover use cases are still valid. CAS buckets must be created with Access During Outage (ADO) enabled, or no data access is possible during temporary site outages.

The ECS CAS implementation is transparent for applications. The Centera HPP protocol stack was ported to ECS and exists inside the data head service along with all other access protocols. A native Centera application can talk to either Centera or ECS without any modifications.

On ECS, the top-level construct is namespace. Multiple namespaces can be created and each can be used by different users for multi-tenancy. There are no special nodes required for access. All nodes have full system view of data and metadata. A Centera cluster is equivalent to a namespace on ECS. Buckets are created inside namespaces on ECS and are similar to pools in Centera. Object users and groups are given access to buckets in ECS and map to profiles in Centera. ECS has a rich set of Access Control Lists (ACL) which allow administrators to grant access for object users and groups to buckets. These ECS ACLs map to Centera's capabilities and are READ, READ_ACL, WRITE, WRITE_ACL, EXECUTE, FULL_CONTROL, PRIVILEGED_WRITE, DELETE, NONE.

Existing Centera SDK versions can handle the same clips, BLOBs and reflections (used to store audit trail of each CDF that is deleted) in ECS because they are stored using the traditional Centera binary format. For more information on ECS see the architectural guide here: http://www.emc.com/collateral/white-papers/h14071-ecs-architectural-guide-wp.pdf.

## ECS SUPPORT FOR CAS COMPLIANCE

Compliance mode is designated at the namespace level in ECS, as opposed to the cluster level on Centera, and ECS does not have the equivalent of Centera's basic compliance mode. On ECS compliance is either enabled or disabled. Compliance can be enabled during namespace creation only so an existing compliance-disabled namespace cannot be upgraded to a compliance-enabled namespace. When compliance is disabled, the ECS system emulates the GE mode in Centera. In other words, retention is enforced but retention can be shortened or lengthened and overridden using privileged delete. When compliance is enabled, the ECS system emulates the CE+ compliance mode in Centera. For compliance-enabled namespaces, retention can be lengthened only and cannot be overridden using a privileged delete. When configured in compliance-enabled mode, it disables privileged deletes and forces bucket retention to be a non-zero value.

There are three ways to define retention on ECS: retention period, retention policy, and bucket retention. Note that there is no default bucket retention in ECS. The default retention the server reports is always 0. Retention period in ECS is used just the way it is used in Centera. It is a simple mechanism for defining retention periods for a clip. Retention policy is an analogue of the retention classes in Centera. Retention policy is configured at the namespace level but applied at the clip level. A retention policy cannot be renamed or deleted. Retention policies provide a way to manage and change retention periods for a set of clips.

Bucket retention provides a way to manage and change retention periods for all clips within the bucket. Note that bucket retention is applied to any clip within the bucket but it is not applied at the clip level. Bucket retention is not a part of clip metadata, however, it is checked for every clip before deleting. Clips with no retention information are protected by bucket retention only. Zero retention is considered a valid value for all retention types except bucket retention for compliance-enabled namespace. Administrators need to specify a non-zero retention for all buckets within compliance-enabled namespaces. Retention also can be infinite (or simply -1). Objects under infinite retention can never be deleted for compliance-enabled namespaces. It is important to understand that applications may experience an error if bucket level retention is set longer than an object level retention. This is because an application may not expect a retention period applied beyond that which is available in object metadata.

Advanced Retention Management (ARM) support is available beginning in ECS 3.0. ARM features are Event Based Retention (EBR), Litigation Hold (LH) and the Min/Max Governor which sets the minimum/maximum bounds for a retention period. Single instancing is not supported on ECS, however, the ECS CAS head assures single instancing at the bucket level when data is copied via ECS Sync or the transformation engine.

## ECS CAS NAMING

The Cluster ID reported by ECS CAS to applications is generated using the ECS software license ID. As a result, if one ECS cluster hosts two or more CAS clusters (namespaces on ECS), all the CAS clusters will have the same cluster ID. This limitation follows from the Centera handshake protocol. Cluster names are reported by ECS CAS to applications that are generated from the cluster ID. Cluster name is {cluster ID}-{namespace name}. Namespace names are unique and immutable in ECS. Bucket name is specified by administrators during bucket creation and they are immutable in ECS. Bucket ID is generated using cluster name and bucket name. Bucket ID is {cluster name}-{bucket name}, for example, e9f1a3c9-5cb1-3a76-a589-f639de9ad2e4-mynamespace-mybucket. Note that for migrated CAS data the bucket ID related consistency is broken and Pool ID within a CDF does not equal to the ID of the bucket it is stored in.

## ECS CAS INTERNALS

CAS clips, BLOBs and reflections are independent objects from the ECS point of view and as such they are stored as native ECS objects. It is the function of the CAS head to unite clips and referenced BLOBs to C-Clips. Currently the CAS head does not delete a clip object and does not create a reflection object with the same content address. Instead, CAS head uses the fact that ECS is a mutable content storage, and overrides clip object content with content of the reflection. There is a specific flag in the head metadata that indicates the object type (clip or reflection). Currently there is no garbage collection of CAS data on ECS, as on Centera, but reflection expiration is supported.

In ECS indexes, directory tables are used to keep system, head, and user metadata for the CAS Query implementation. A single ECS cluster may contain several CAS clusters so the CAS head does the following two steps in order to derive the target bucket for connection. First, the CAS head uses the user (profile) name to derive the target namespace, and second, the namespace/user pair is used to derive the target bucket. There is a limitation that a CAS user must have a unique name within ECS cluster.

In Centera, a cluster is a de-facto global namespace. There is a global index of unique objects (clips, BLOBs, and reflections). There is a so called cluster pool that unites all the objects within the system. This approach allows a single user (profile) to read, query, delete, etc., C-Clips from different application pools. ECS uses a different approach. In ECS an index of unique objects is maintained at the bucket level which is why a user can perform read, query, delete, etc. against its home pool only. The optional default bucket, which can be specified for each user, stands for home pool for CAS applications. ECS has no predefined buckets like default or system pools on Centera.

CAS applications can work with CAS-enabled buckets only. Content of CAS-enabled buckets is only accessible using the CAS interface. Therefore, no backdoor access is possible. When a connection to CAS is set up, the server uses the user name provided to discover the target pool/bucket. That is why CAS supports unique user names only. ECS configuration with several namespaces with users of the same name is not supported.

### CORE DIFFERENCES BETWEEN CENTERA AND ECS

- Where Centera is limited to use only by CAS applications, ECS serves all major object protocols: Amazon S3, OpenStack Swift, Dell EMC Atmos, and Dell EMC CAS. In addition, ECS supports NFSv3 and is able to serve contents directly to a Hadoop compute cluster either as the primary or a secondary Hadoop Compatible File System (HCFS).

- ECS provides a global namespace that can be spread across up to eight geographically distributed locations. The active/active geo-topology is built on a strongly consistent data model which has many benefits over eventually consistent replication models such as with Centera. Global access is available for all supported protocols so NFSv3 data written to one location, for example, can be read on another ECS system in an entirely different location, and, analytics can be performed on data located across all sites.

- ECS provides HTTP(S) REST APIs for modern day application development so while traditional CAS applications can use ECS as a direct replacement for Centera, the same ECS system can simultaneously serve modern applications for a true multi-purpose storage solution.

- ECS offers a lower Total Cost of Ownership (TCO) when comparing a three year extension of maintenance contract versus new ECS with warranty. ECS is highly dense and consumes less power per terabyte. Small objects force underutilization of capacity on Centera. ECS is optimized for both small and large objects and its 12+4 and 10+2 erasure coding schemes lead to better capacity utilization.

- Centera is constrained by a maximum object count of one hundred fifty million objects per node with CentraStar 4.3 and later versions. Earlier versions of CentraStar are constrained to 100 million objects. Once a Centera system reaches its object limit it becomes unavailable to new content until existing object are removed. ECS allows CAS applications to bypass this limitation by providing a system with no ceiling for object count.

- On Centera, Dell EMC recommends a maximum BLOB size of no more than 100 GB and CDF size is capped at 100 MB. CAS on ECS does not have these same limitations.

### MIGRATION REQUIREMENTS

1. **Verify application compatibility.** Application compatibility with ECS needs to be verified. Data Shredding is not available as of ECS 3.0.

2. **Determine which applications will be moved to ECS.** Applications requiring access to data that is moved to ECS must be cutover and also moved to ECS. It is not possible for an application to access data that was moved to ECS by continuing to connect directly to Centera. It is possible however, due to the new transformation component, for an application to access data on Centera by connecting directly to ECS.

3. **Determine which data will be moved to ECS.** In general, all data associated with applications that are moved to ECS will also be moved to ECS. The transformation engine allows ECS to serve as a front-end to Centera storage, that is, for the two systems to co-exist. This co-existence is possible because ECS is able to learn about all of the data on Centera, access it upon request by an application, and serve it to application(s). New data will be written on ECS.

4. **Determine if Dell EMC Professional Services (PS) is required or desired.** ECS Sync can be used by customers without the need for a Dell EMC PS engagement. Data migration using the transformation engine is initially available via Dell EMC PS engagements only. PS can provide a full-service complete application and data migration using either tool. NOTE: ECS Sync cannot handle migration of EBR/LH object information.

## ECS SYNC

ECS Sync was written specifically to move large amounts of data while maintaining application association and metadata. With ECS Sync, you can not only migrate C-Clips from Centera to ECS but you can also pull BLOBs out of a database and move them into an S3 bucket. It is a multi-purpose, multithreaded tool with a set of plug-ins that speak native protocols. ECS Sync queries the source system for objects using CLI or XML-configured parameters. It then streams these objects and their metadata in parallel across the network, transforming/logging them through filters, and writing them to the target system, updating database references on success.

Encryption, data verification as well as tracking and saving progress are integrated in to the tool and recently both CLI and web-based UI are available as separate downloads. Using ECS Sync is similar to most common data migrations between disparate systems where he data is moved and applications are cutover.

At a high level, data migration with ECS Sync is as follows. ECS Sync is installed on one or more migration hosts along with its required dependencies, Java 1.7, MySQL, and the CAS SDK. Network connectivity is required and verified between the host(s) and Centera and the host(s) and ECS over ports 3218 and 22. The IP addresses along with necessary credentials and PEA files are collected.

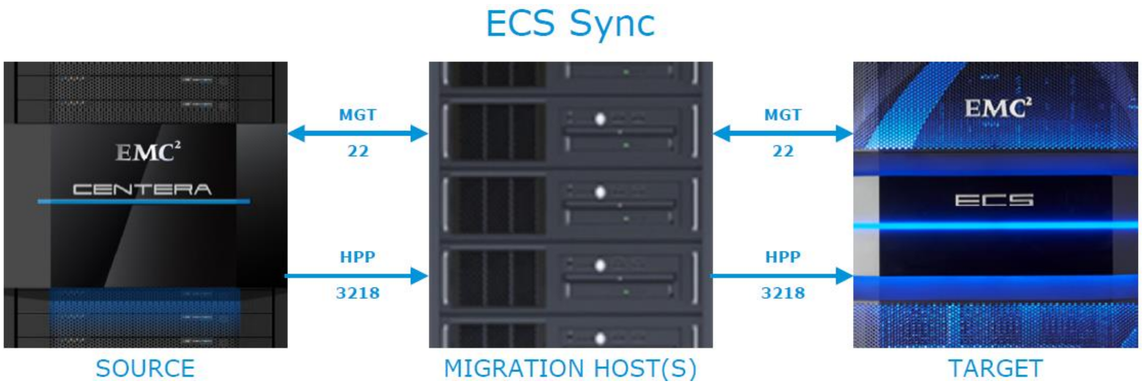

Figure 1

A file containing all CAS Clip IDs requiring migration is required and is referred to as the "clip list." The clip list can be created using ECS Sync to query Centera, but any method can be used, for example, a query to an existing data base which holds the required information. With the clip list a CLI command can be generated that specifies the clip list, source Centera, target ECS and any optional filters, to move the data.

All data being moved is first copied in full to the target system and, if needed, incremental changes are copied afterwards until the scheduled application cutover. Just before application cutover, updates to the source data need to be prevented. This can be accomplished either by shutting down the application(s) or making the data read-only. Once writes are stopped on the source, any changes not yet migrated are copied to ensure both sets of data are identical. Next, post-migration data validation is completed and any conflicts between the two data sets need to be resolved. Applications are then pointed to the ECS and tested for proper access.

As with traditional migrations of this type, application downtime is, at minimum, the amount of time to complete a final incremental sync of data, validate that the data between both systems are identical, and start the application pointed to the new system. Dell EMC personnel generally see an average of 1.5 million clips moved per day, per migration host, for objects less than 250 KB, and 2 TB of data moved per day for objects greater than 1MB. Data validation is included in these rates. Results vary based on the size and latency of network links. Experience has shown that a maximum of one migration host per two Centera access nodes is ideal.

Single instancing is not supported by ECS Sync. This means if a single BLOB on Centera has one hundred links to it, the BLOB will be copied to ECS one hundred times. However, single instancing at the bucket level will be assured by ECS, if it runs ECS 2.2.1 HF1, and later. For example, if a BLOB is referenced by 100 CDFs from 10 different source pools in Centera, at the end ECS will have only 10 copies of the BLOB, i.e. one copy per target ECS bucket. The latest version of ECS Sync is available on GitHub at https://github.com/EMCECS/ecs-sync.

## ECS TRANSFORMATION ENGINE

The ECS transformation engine was developed to allow customers to seamlessly move applications from Centera to ECS without application logic modifications and without significant service interruption. The transformation engine, part of the data services layer responsible for all application/client-side protocol communication, allows ECS to both operate alongside Centera, serving applications data that physically resides on Centera, and, optionally migrate data from Centera to ECS.

The transformation engine performs two primary functions:

- Transformation - ECS serves data stored on Centera to applications, and transforms the data in to native ECS objects.
- Migration - data is physically copied to ECS and reconciled for accuracy.

It is possible to only run the transformation-related steps on Centera data which will allow applications to move to ECS but not their data. After transformation of data however it is additionally then possible to trigger migration of data to ECS with the intent of decommissioning the application/workflow off Centera. It is not possible to migrate data without first transforming it because it is the transformation process that enables ECS to both serve and store Centera data using its own constructs. The transformation of data is where ECS maps a Centera cluster to an ECS namespace, pools to buckets, profiles to users, and capabilities to ACLs.

Application downtime, when using the transformation engine, can be more seamless as compared to when using ECS Sync. This is because the downtime for a cutover is purely dependent upon time for an application to restart or reload new connection parameters.

The data transformation component supports Centera CentraStar 4.0 or later firmware on Gen4 or Gen4LP hardware. A federated Centera environment, where a collection of Centera clusters form a virtual archive, is not supported, nor are Centera implementations with multiple replicas. Single instancing is not supported by ECS but is preserved when data is copied using the transformation and migration feature. This means if a BLOB on Centera has one hundred links to it, the BLOB will be copied to ECS once for each bucket and the related C-Clips will each have a pointer to it. In general, 4-5 TB of data can be moved per four ECS nodes in day, however, as with ECS Sync results easily vary and are based on factors such as system load, number of access nodes, and sizes of objects.

## TRANSFORMATION

Transformation requires connectivity between Centera and ECS over ports 3218 for data and 3682 for management access. The transformation engine handles both the transformation of data as well as its migration. Migrated data must first be transformed. Migration isn't required and the process can be stopped after transformation if it is desired for the data to remain on Centera.

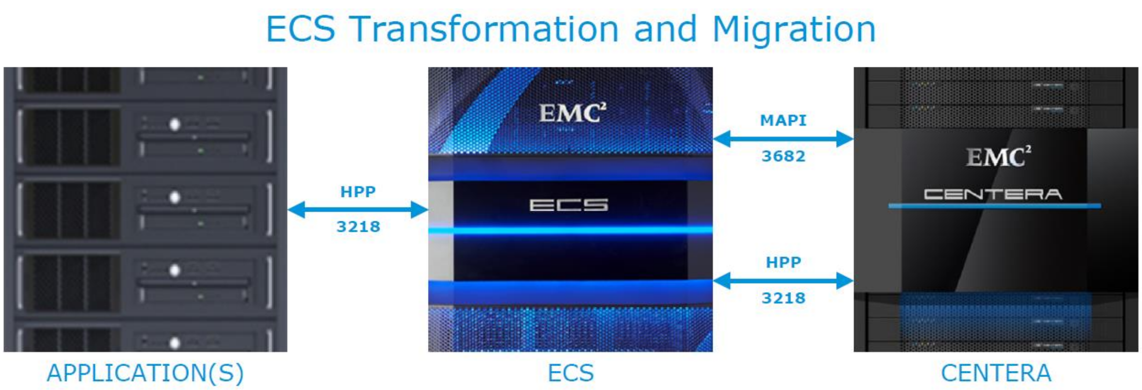


Figure 2

The ECS transformation engine is what allows ECS to serve as the middle-man to data on Centera storage and it allows for the number of objects and quantity of storage to limitlessly increase without abandoning the Centera system. Using the transformation engine, data for applications which are compatible with ECS CAS can be transformed and even migrated if desired, while applications requiring not-yet-implemented features such as Data Shredding, can remain untouched on Centera.

| ECS PERMISSION | CENTERA MASK | CENTERA CAPABILITIES |
|---|---|---|
| Read | r-qe---- | Read, query, exist |
| Write | ------wh | Write, Litigation Hold |
| FULL_CONTROL | rdqe-cwh | Read, Delete, Query, Exist, Clip copy, Write, Litigation hold |
| PRIVILEGED_WRITE | ----D--- | Privileged delete |
| DELETE | -d------ | Delete |

Table 3

It is the transformation process that educates ECS on data location and information on Centera. As part of the transformation of data, a mapping is created between Centera and ECS constructs. A Centera cluster maps to an ECS namespace, a Centera pool maps to an ECS bucket, a Centera profile to an ECS user and Centera capabilities to ECS ACLs. CAS users are generic ECS object users with optional properties defined. These properties are the CAS secret key and default bucket. A CAS user can access only one bucket at a time but a bucket could be accessed by multiple users.

The transformation engine maps ECS ACLs to Centera capabilities as per Table 3. Since there isn't a one-to-one mapping of ACLs to capabilities, some applications may use a Centera mask that can't be reproduced exactly on ECS.

Newly written data is stored in ECS and all application write requests (including updates and deletes) have no effect on Centera and are handled only by ECS. Application read requests are propagated to Centera transparently for applications if an object cannot be found in ECS. If a read request is found on ECS, as a result of an object being migrated, it will be served from ECS.

Transformation (and migration) processes are executed only by ECS nodes in a single Virtual Data Center (VDC) and are not accessible globally. That is, all other VDCs in the geo ensemble have no notion about the transformation and so the processes must be managed via the local-to-Centera ECS. If the local-to-Centera ECS performing the transformation becomes permanently unavailable, the transformation process must be started fresh on a different ECS system.

All users, buckets, and indexed objects are available through every VDC within a replication group. However, because indexed objects remain in Centera, they are not replicated across all VDCs in replication group. That means that all reads of indexed objects directed to a remote VDC are redirected to the VDC hosting the transformation. In other words, accessing indexed objects through remote VDCs always causes inter-VDC traffic. All data on Centera will be unavailable if the Centera or ECS cluster performing the transformation become inaccessible.

You cannot create or execute transformations unless all ECS nodes in a VDC are upgraded to a version including the transformation feature.

## TRANSFORMATION WITH MIGRATION

Once transformation has been completed, the optional migration steps can begin. Migrating data using the data transformation component is done in the background by ECS with prioritization of resources for production traffic. All nodes participate in the movement of data. ECS erasure codes data while it is being copied.

During a migration, which is based on creation date, since objects will have creation date on ECS represented by time of data migration to the ECS system, ECS honors the original write date on the Centera system for replication purposes.

## TRANSFORMATION AND MIGRATION PHASES

The transformation engine is responsible for all phases of the transformation and migration. The complete transformation and migration lifecycle consists of eight phases. These are configuration, pre-check, enumeration, indexing, migration, reconciliation, re-migration, and deleting, as seen below in Figure 3. Transformation is handled by the first four phases and the last four are for migration.

Each phase can only end successfully and as such any non-successful outcomes do not count as a phase end. Possible phase outcomes are failed, cancelled and succeeded. Progress is maintained to ensure failed and cancelled phases can be retried from where they left off. Each phase has an associated job and most jobs are long-running background processes which are carried out by all nodes concurrently via a set of job tasks. Basically each job is an internal representation of a phase which defines the amount of work to be done.

Phases are executed in strict order and each individual phase is executed by all ECS cluster members concurrently. Processes are controlled by entering commands on the ECS CLI, via the API or directly via the WebUI as of ECS 3.0.

### PHASE 1: CONFIGURATION

The process begins with the creation of a transformation instance on ECS which allows for access to a Centera cluster. Configuration only requires a few key Centera variables such as a management IP, credentials for the Management API (MAPI) user, and the ID of an existing replication group on ECS for use in the transformation. During the creation of the transformation instance a transformation ID and new namespace name are generated and all of the pools and related profiles are retrieved. The transformation ID is used as a key parameter in all subsequent phases. It is not required to transform all cluster data so prior to initiating the pre-check, the list of pools to be transformed ("transformation sources") can be edited along with the auto-generated names for the namespace and buckets.

### PHASE 2: PRE-CHECK

During phase two ECS first performs several health checks to ensure the system is available for transformation. Issues discovered during the pre-check phase are included in the phase report and should be reviewed, corrected, and if needed the pre-check phase should be run again, until a successful status is returned.

Descriptions of the pre-checks are:

- Provisioned Centera MAPI user has enough privileges

- Centera reports data integrity incompleteness
- Centera hardware version does not met requirements
- Centera upgrade is in progress
- Centera software version does not met requirements
- Unsupported compliance mode
- There are still unresolved Profile Mapping conflicts
- No source pools are added to the Transformation
- Configured source pools is not accessible by profile given

After the pre-checks complete successfully, provisioning of namespace, buckets and users begins and ECS verifies source data can be transformed. Descriptions of the provisioning-related checks are:

- Ensures that target namespace for transformation buckets exists, namespace will be created if missing. New namespace admin user will be created automatically for newly created namespace
- Ensures that target buckets exist, buckets will be created if missing
- Ensures that ECS users exist, users will be created if missing
- Ensures that ECS CAS retention classes match Centera, ECS CAS retention policies will be created if missing
- Ensures that configured Centera compliance mode is stored in Transformation metadata
- Ensures that miscellaneous data user accounts needed at reconciliation phase are created

After all checks are complete, application(s) need to be cutover to ECS using PEA files generated on ECS. The application cutover can be done with minimal disruption and any downtime is purely limited to the amount of time it takes for an application to be pointed from Centera to ECS. It is important to note that the transformation engine does not have any rollback feature. Any writes by an application that has been cutover to ECS will reside on ECS and not the Centera. If it desired for the application to be cut back to the Centera, and the application has written or modified content, manual intervention will be required to ensure the Centera cluster pool has the new data that was written to ECS.

After the pre-check phase is complete, the two systems are said to coexist. Coexistence is where ECS serves data that exists on Centera, and all new application writes are stored on ECS, essentially both systems remain online and active.

**PHASE 3: ENUMERATION**

During the enumeration phase ECS creates a set of Content-Address objects which represent all content stored in the pools being transformed. This is done using a Centera query where ECS uses the write time of the first written clip on Centera for the lower bound and current time as the upper bound. ECS initially chops this time in twenty-four hour periods and splits these cross sections of work across all nodes in order to parallelize processing. The enumeration of different time frames is split without strict ordering. Since ECS doesn't know how densely clips are located on time some enumeration tasks may take longer than others. Work may be broken down in to smaller chunks of time if a specific period is found to be taking too long.

The enumeration phase report includes all periods of time for which the query returned incomplete results. Due to state of Centera, like if it needs healing for example, enumeration may need to be retried.

Again, it is important that applications lose access to their pools on Centera and are cutover to ECS once the pre-check phase is completed. Enumeration should not be started if an application (moving to ECS) is still using Centera for storage. It is safe to proceed with the enumeration phase only when application(s) have been cutover to ECS. Otherwise there is no guarantee that ECS will discover all Centera content correctly.

**PHASE 4: INDEXING**

It is during the indexing phase that ECS processes the previously created list of C-Clips and BLOBs and builds internal indexes of the Centera content. Various internal directory tables are used to store information about indexed objects. Once an object is indexed it becomes indistinguishable from any other ECS object and becomes available globally through all ECS VDC members in the replication group used during the transformation. At this time indexed objects are still located on Centera. After successful completion of the indexing phase all of the source data is considered transformed.

No space is reserved in ECS for object migration during the indexing phase. Quota space is checked during indexing for quota-enabled namespaces and buckets, but, new ingest coming to ECS in parallel with migration can consume space needed for the migrated objects placement and if insufficient space becomes available a migration can fail.

Indexing also prepares for the migration by pre-creating chunks and planning in advance how objects will be packed in to them.

**PHASE 5: MIGRATION**

The migration phase, along with the remaining phases, is only necessary if the transformed data will be migrated to ECS. During the migration phase all of the chunks created during the indexing phase are filled with the transformed Centera content. The content is erasure coded as it is copied and due to the concurrent and orderless nature of migration some objects may be partially migrated at any particular point in time.

**PHASE 6: RECONCILIATION**

Centera object hash code (CRC) is used to verify migrated object's consistency. If no hash code is provided by Centera additional object read requests will be issue by ECS. This phase ensures all data requiring migration was actually migrated and can be read as well as verifying correct Centera compliance-related information.

Reconciliation uses special ECS data users which have no mapping to Centera profiles. This ensures that ECS logic won't fallback to Centera while reading objects. The users (user: "ra") are created in the namespace during the pre-check phase.

**PHASE 7: RE-MIGRATION**

Any issues found during reconciliation are fixed during re-migration. Issues are derived from a list of incorrectly migrated objects, if any, built during the reconciliation phase. The list is populated with objects which are not readable from ECS or their checksum does not match. No automatic re-migration is triggered. Re-migration is triggered manually for all items in the list in one fell swoop. Re-migration of individual objects is not supported. The process of reconciliation should be triggered again, post re-migration, to verify that objects were re-migrated correctly.

**PHASE 8: DELETING**

During this phase all transformation-related artifacts, such as phase reports, are removed from the system, including the transformation instance. At the time of this paper's creation, the one exception to this is for the "ra" user created in the namespace to verify all data can be read successfully. User "ra" requires manual deletion.

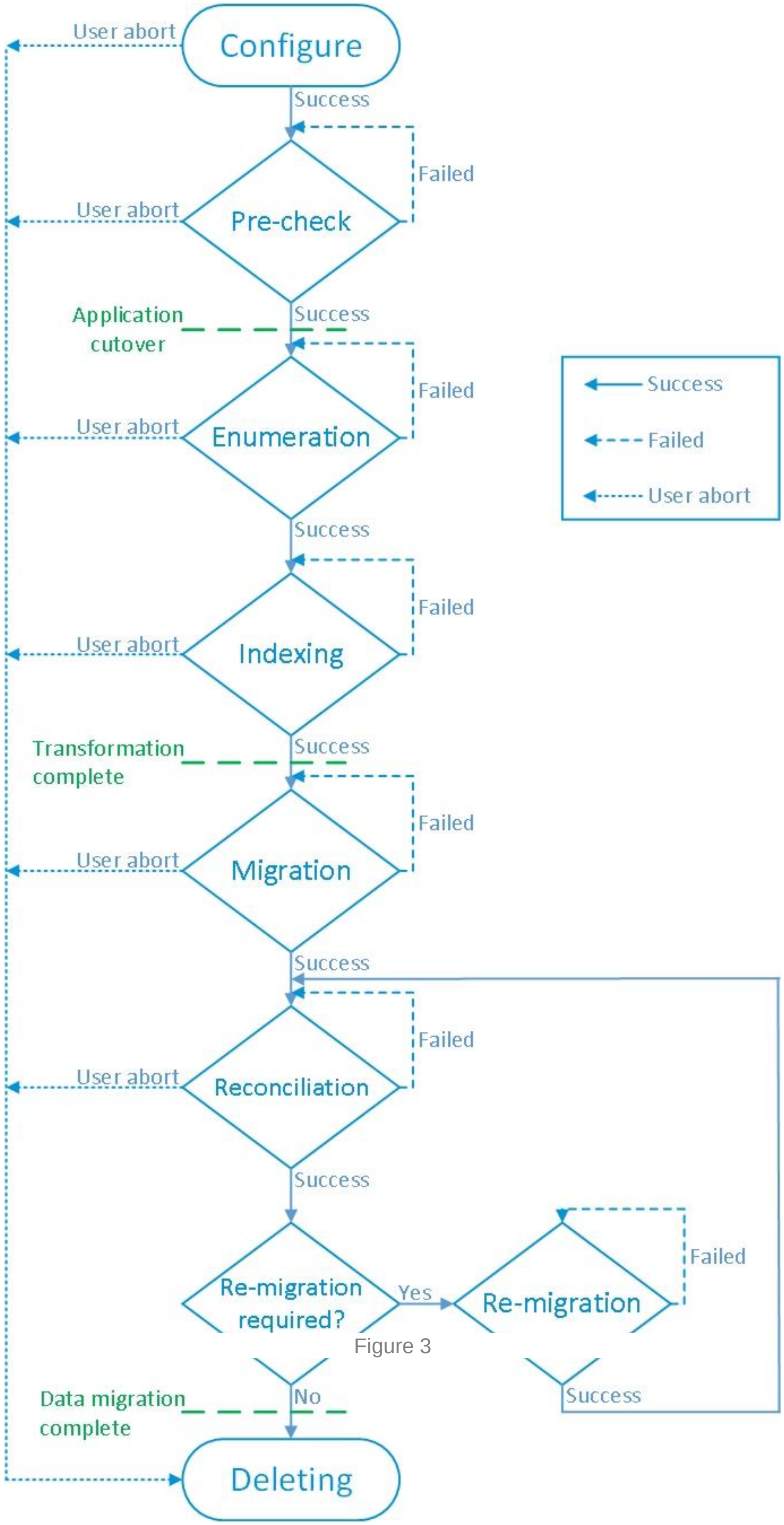
Configure
User abort
Success
Pre-check
Failed
Application cutover
Enumeration
Indexing
Transformation complete
Migration
Reconciliation
Re-migration required?
Yes
No
Re-migration
Data migration complete
Deleting
Success
Failed
User abort

Figure 3

## CONSIDERATIONS

- ARM features are available for CAS only.

  A Maximum limit of 100 Litigation Holds per object exists. The current number of Litigation Holds per object cannot be queried, however, a query is available for Litigation Hold and returns True, if one or more holds exist, or False if none.

- There can only be one transformation instance for each Centera cluster at a given time.

  Several Centera clusters can be transformed and migrated concurrently but at any given time there can only be one transformation instance for each Centera cluster. This is important to consider when only a portion of the data to be transformed on a cluster will be migrated. To briefly review, transformed data is data that exists on Centera but is served to an application by ECS. The first step to migrating data to ECS is transforming it, but, migration of data isn't required post-transformation. For a given transformation instance, once the transformation has been completed, if migration is desired, all of the transformed data must be migrated, that is, it isn't possible to only migrate a portion of the transformed data. In scenarios such as this, where applications will be cutover to ECS but only some of their data will remain on Centera, more than one transformation instance is required.

  If one or more transformation instances are required for a Centera cluster, the first instance(s) should be created to accomplish data migration. After successful migration of data the related transformation instance is deleted and a new instance can be created for the same cluster. The last transformation instance created will be for the non-migration transformation-only data. This transformation instance will not be deleted and remains on the system for as long as the two systems need to co-exist. ECS will store new application data and serve application(s) data that remains stored on the Centera cluster.

- The transformation engine cannot move data from ECS to Centera.

  It is possible for a transformation instance to be deleted at any time. Similarly applications can be cut back over to Centera at any time. If however an application is moved from ECS back to Centera, any new writes or modifications that occurred while the application was connected to ECS will only be available on ECS. Manual intervention will be required in order to copy the newly created data from ECS to Centera. At no time does ECS delete any data on Centera and by default ECS connects to Centera in a read-only fashion. ECS Sync can copy data from ECS to Centera and is an option in this scenario.

- Applications need to be cutover to ECS after the pre-check phase and before the enumeration phase.

  When the enumeration phase starts a Centera query is built using the current time for one of the bounds. If any writes occur in transformed pools on Centera after that point in time ECS will be unaware of them and they will not be migrated.

- Transformed data is only available globally after it has been indexed.

  Once data has been indexed it is considered an object on ECS just like all other objects. As such if the replication group spans more than one VDC the indexed data, which remains on Centera, is accessible from other VDCs. Until an object is indexed it is only available locally to the ECS responsible for transforming the data.

- Data migration will fail if a related quota is exceeded.

  For quota-enabled namespace, a check is performed to ensure sufficient space is available to accommodate the data being migrated. The check is performed at the start of the indexing phase however so it is possible for non-transformation-related object creation to push the capacity beyond its allowed limit causing the migration to fail.

- If VLAN separation is used, all migration traffic must be configured to use the data VLAN.

- Applications which push Centera close to the edge of its performance limits, and consist of multithreaded peak performance smaller object workloads or single-threaded read workflows, should refer to the ECS CAS API Performance Whitepaper (http://www.emc.com/collateral/white-papers/h14927-ecs-cas-api-performance-wp.pdf), to ensure application needs can be met with ECS.

- It is highly recommended to test data migration prior to implementation.

# SUMMARY

Dell EMC is committed to Centera however applications and their underlying storage needs have advanced rapidly since its inception. ECS was built from the ground up to be the next modern-day unstructured storage platform. Although organizations may initially only be considering ECS due to the need to replace EOSL hardware, to add additional capacity, or because of pending maintenance renewals, when looking at everything ECS offers, along with the ease of transition from to Centera to ECS, it is likely moving applications to ECS makes the most sense.

With the exception of Data Shredding and Single-Instancing, CAS applications can use ECS today without any need for modifications. By using the standalone ECS Sync application or ECS's Centera Data Transformation and Migration feature, data and their applications can easily be moved from Centera to ECS. By transitioning to ECS organizations can open the door for access to storage via all major object protocols as well as NFS and HDFS. In addition, on ECS, existing limitations on a Centera cluster such as a limit to the total number of objects allowed and even object size, become mute. ECS is scalable to a global level and with its active/active, read/write everywhere strongly consistent data model, the failover/failback replication scenarios on the storage side can easily become a thing of the past.

At its most basic Centera consists of clusters, pools, profiles and capabilities. Similarly ECS has namespaces, buckets, users, and ACLs. By transforming data the Centera constructs are mapped to ECS constructs and applications are able to access existing data by connecting directly to ECS. An option to the retirement of a Centera cluster is the possibility for the two systems to co-exist, and for ECS to offload some or all of the front-end application connectivity for Centera, while all or some data continues to remain on Centera. This may be useful to bypass Centera total object count limitation or allow for access from multiple locations.

There are two primary decisions when considering data migrations to ECS:

- Application compatibility
- The migration method

ECS Sync might make more sense, especially when longer application downtime can be tolerated. Natively with ECS's transformation engine seamless integration with often trivial downtime may be more desirable and at the time of publishing it's the only option to migrate objects with EBR and LH object information.

# APPENDIX A - CREATING A CAS BUCKET ON ECS

## CREATING A CAS BUCKET ON ECS

Along with the discussion of moving from Centera to ECS, some may find it helpful to understand the configuration required on ECS to support a CAS application, independent of any migration. Using the WebUI it is very simple to provision ECS for CAS access. A new or existing namespace can be used and can optionally have one or more retention periods set. A user is created for the application in the namespace and then a bucket is created using the new user as the bucket owner, with or without a retention period. A 'Bucket Retention' section provides ARM-related options. Admin can enable enforcement of retention information in object for the bucket, a bucket retention period, and Min/Max periods for both fixed and variable retention periods as with EBR.

After bucket creation, the bucket owner can then be edited to include CAS-specific information. In the CAS section towards the bottom of the 'User Management' page for the user, a password should be entered manually or generated using the 'Generate' button. Afterwards the 'Set Password' button will become available and when pressed reveal the 'Default Bucket' selection option. Here the newly created bucket for the application is selected and solidified when the 'Set Bucket' button is pressed. A PEA file can be generated on this page for use by the application for bucket access and attributes can be added to the user at that time as well. Using the PEA file in a Centera-compatible application, the application can now be pointed to the ECS cluster and will have write access to the bucket created.