Configuration and Deployment

Dell EMC ECS: Splunk SmartStore Configuration Guide

Abstract

This document describes how to configure Splunk® SmartStore with Dell EMC™ ECS.

June 2019

H17780.1

Revisions

Date	Description
June 2019	Initial release
March 2021	Dell Technologies template updates

Acknowledgments

This paper is produced by the Dell EMC Unstructured Technical Marketing Engineering and Solution Architects team.

Author: Rich Paulson, Kirankumar Bhusanurmath.

Table of contents

Executive summary

The explosive growth of unstructured data and cloud-native applications has created demand for scalable on-premises cloud storage infrastructure in the modern data center. Dell EMC™ ECS, the third generation of object store by Dell EMC, is designed from the ground up to take advantage of modern cloud storage APIs and distributed data protection, providing active/active or active/passive availability spanning multiple data centers.

As data volumes continue to increase in both capacity and long-term business analytics value, Splunk® SmartStore indexes and raw data can be stored externally using a cost-effective object store compatible with Amazon® Simple Storage Service (S3), such as ECS. This allows customers to scale their indexer storage resources separately.

This white paper is a reference guide for configuring Splunk SmartStore with ECS and includes recommended best practices.

Audience

This document is intended for administrators who deploy and configure Dell EMC ECS with Splunk SmartStore. This guide assumes a high level of technical knowledge for the devices and technologies described.

ECS terminology

Replication group: Replication groups are logical constructs that define where data is protected and accessed. Replication groups can be local or global. Local replication groups protect objects within the same VDC against disk or node failures. Global replication groups span two or more federated VDCs and protect objects against disk, node, and site failures.

The strategy for defining replication groups depends on multiple factors including requirements for data resiliency, the cost of storage, and physical compare with logical separation of data. As with storage pools, the minimum number of replication groups required should be implemented. At the core ECS indexing level, each storage pool and replication group pairing is tracked and adds significant overhead. It is best practice to create the absolute minimum number of replication groups required. Generally, there is one replication group for each local VDC, if necessary, and one replication group that contains all sites. Deployments with more than two sites may consider additional replication groups, for example, in scenarios where only a subset of VDCs should participate in data replication, but, this decision should not be made lightly.

Namespace: Namespaces enable ECS to handle multitenant operations. Each tenant is defined by a namespace and a set of users who can store and access objects within that namespace. Namespaces can represent a department within an enterprise, can be created for each unique enterprise or business unit, or can be created for each user. There is no limit to the number of namespaces that can be created from a performance perspective. Time to manage an ECS deployment, on the other hand, or, management overhead, may be a concern in creating and managing many namespaces.

Bucket: Buckets are containers for object data. Each bucket is assigned to one replication group. Namespace users with the appropriate privileges can create buckets and objects within buckets for each object protocol using its API. Buckets can be configured to support NFS and HDFS. Within a namespace, it is possible to use buckets as a way of creating subtenants. For performance reasons, it is not recommended to have more than 1000 buckets per namespace. Generally, a bucket is created per application, workflow, or user.

Object user: Object users are defined by a username and a secret key that can be used to access the object store. Usernames can be local names or can be domain-style username that includes a @ in their name.

Splunk terminology

Bucket: Splunk Enterprise stores indexed data in buckets, which are directories containing both the data and index files into the data. An index typically consists of many buckets, organized by age of the data.

Indexer: A Splunk Enterprise instance that indexes data, transforming raw data into events and placing the results into an index. It also searches the indexed data in response to search requests.

Search: The primary way users navigate data in Splunk Enterprise. You can write a search to retrieve events from an index, use statistical commands to calculate metrics and generate reports, search for specific conditions within a rolling time range window, identify patterns in your data, predict future trends, and so on.

Master node: The indexer cluster node that regulates the functioning of an indexer cluster.

1 Solution overview

This section provides an overview of the integration of Dell EMC ECS with Splunk SmartStore and the key technologies used.

1.1 ECS overview

ECS provides a complete software-defined consistent, indexed, cloud storage platform that supports the storage, manipulation, and analysis of unstructured data on a massive scale. Client access protocols include S3, with additional Dell EMC extensions to the S3 protocol. Object access for S3 is achieved using REST APIs. Objects are written, retrieved, updated, and deleted using HTTP or HTTPS calls using REST verbs such as GET, POST, PUT, DELETE, and HEAD.

ECS was built as a distributed system following the principle of cloud applications. In this model, all hardware nodes provide the core storage services. Without dedicated index or metadata master nodes the system has limitless capacity and scalability.

Service communication ports are a part of the configuration when configuring a SmartStore index to store index data in ECS. See Table 1 for the associated ports used with the ECS S3 data service protocol.

Note: An implementation would use an external IP load balancer to manage traffic flow to the ECS nodes.

Table 1 ECS S3 protocol and associated ports

Protocol	Transfer Protocol	Port
S3	HTTP	9020
	HTTPS	9021

1.2 Splunk SmartStore overview

SmartStore is an indexer capability that provides a way to use remote S3 object stores, such as ECS, to store indexed data.

As a deployment's data volume increases, demand for storage typically outpaces demand for compute resources. SmartStore allows you to manage your indexer storage and compute resources in a cost-effective manner by scaling those resources separately.

SmartStore introduces a remote storage tier and a cache manager. These features allow data to reside either locally on indexers or on ECS. Data movement between the indexer and ECS is managed by the cache manager, which resides on the indexer.

With SmartStore, you can reduce the indexer storage footprint to a minimum and choose I/O optimized compute resources. Most data resides on ECS, while the indexer maintains a local cache that contains a minimal amount of data: hot buckets, copies of warm buckets participating in active or recent searches, and bucket metadata.

SmartStore offers several advantages to the deployment's indexing tier:

- Reduced storage cost. Your deployment can take advantage of the economy of ECS, instead of relying on costly local storage.
- Access to high availability and data resiliency features available through ECS.
- The ability to scale computes and storage resources separately, thus ensuring that you use resources efficiently.
- Simple and flexible configuration with per-index settings.

An intelligent cache manager ensures that, for most search use cases, SmartStore provides similar performance to local storage configurations.

1.3 Solution architecture

Figure 1 shows the workflow as event data is ingested, indexed, and uploaded or downloaded to ECS. When a hot bucket on the local storage cache reaches a certain size, that bucket is uploaded to ECS (warm). If a search includes data that is stored in ECS, that bucket data is downloaded to the local storage cache and in search results. In this example, there are two federated ECS sites. Data is replicated between sites so if there is a site outage, data can still be accessed.

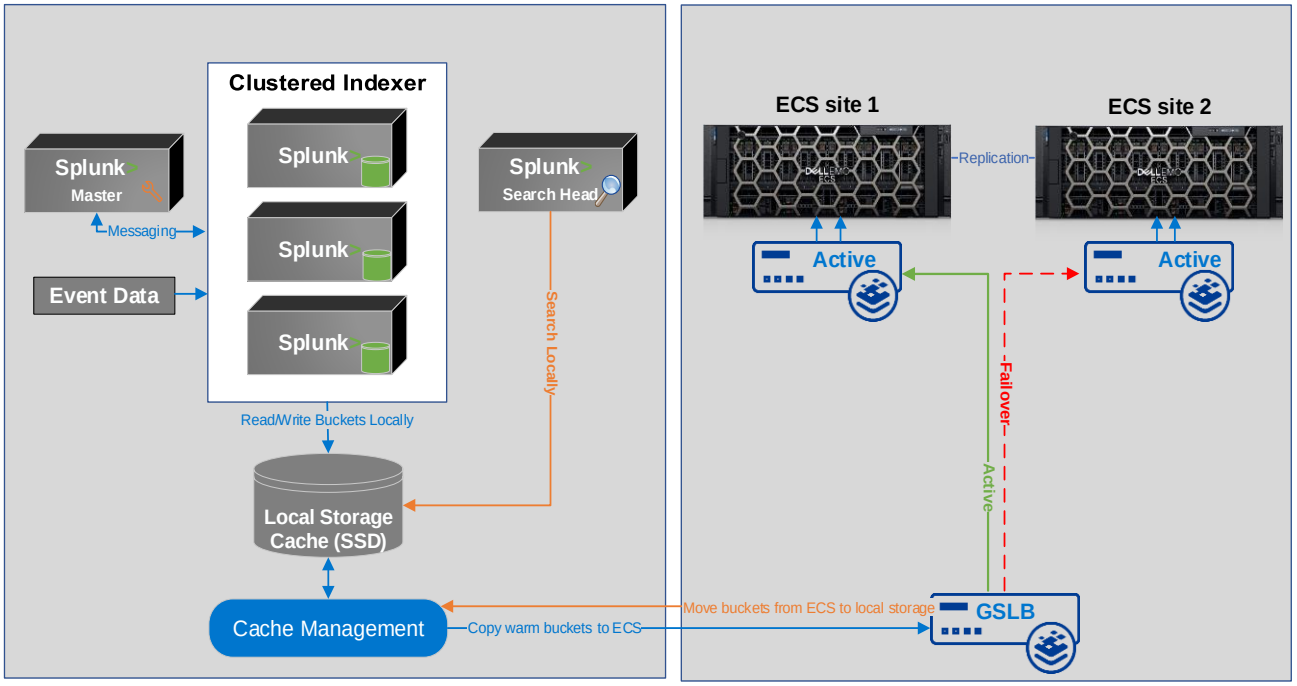

Figure 1 Architectural workflow

1.4 Key components

This section describes the Dell EMC integration components.

Table 2 Del EMC components

Component	Description
ECS	Version 3.3

Table 3 Splunk components

Component	Description
Splunk Enterprise	Version 7.3

2 Solution implementation

This section describes the steps required to configure a Splunk SmartStore index with Dell EMC ECS.

2.1 Implementation workflow

The following are the minimal steps required to implement the solution.

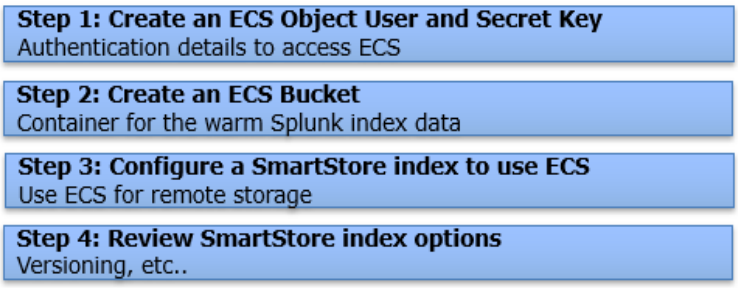

Figure 2 Configuration steps

2.1.1 Create the ECS object user, secret key, and bucket

An object user, S3 secret key, and bucket will need to be created for SmartStore to access ECS to store index data. This can be done from the ECS Web Portal or Management API.

1. Create an object user and generate an S3 secret key.

   a. From the ECS Web Portal, go to **Manage** > **Users** and click the **New Object User** button.
   b. Enter the name of the user and a namespace.

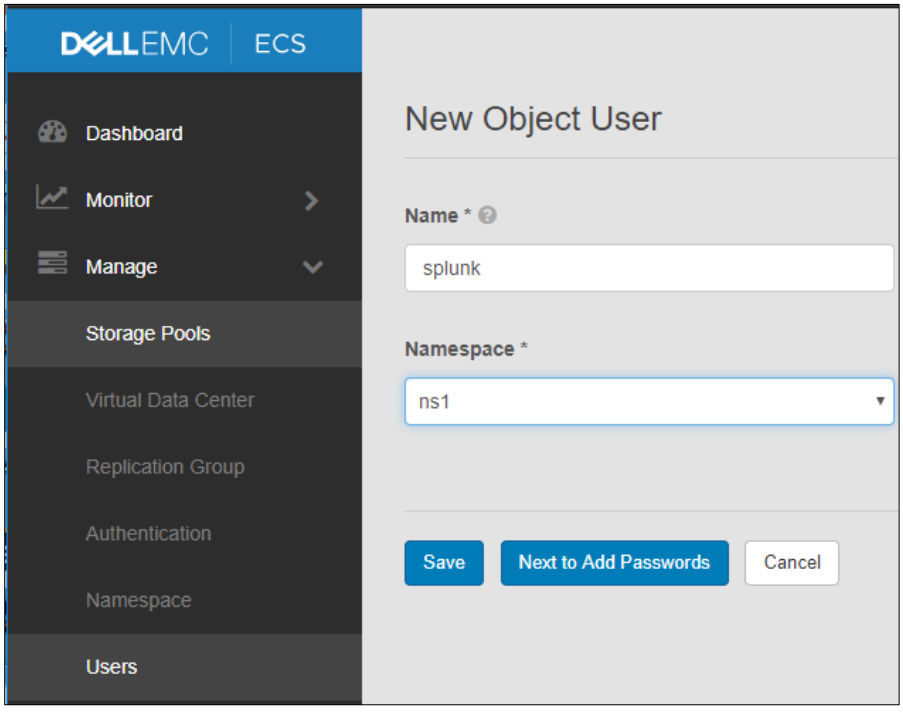

Figure 3 Create an ECS object user.

c. Click the **Next to Add Passwords** button and click the **Generate and Add Secret Key** button under the **Object Access** section.

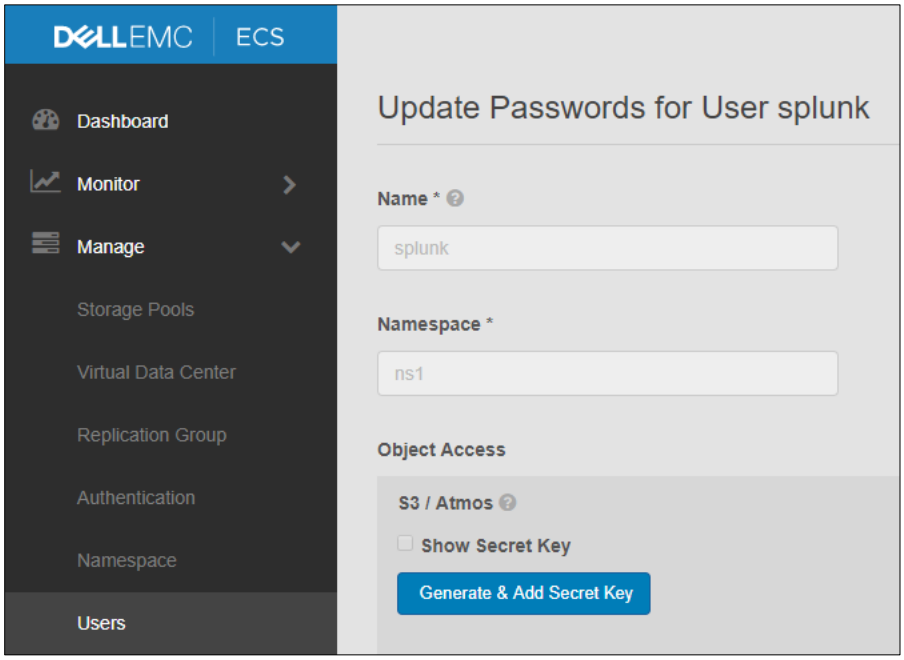

Figure 4 Generate an S3 secret key.

d. Click the **Show Secret Key** box to display the secret key. Click the **Close** button at the bottom of the page.

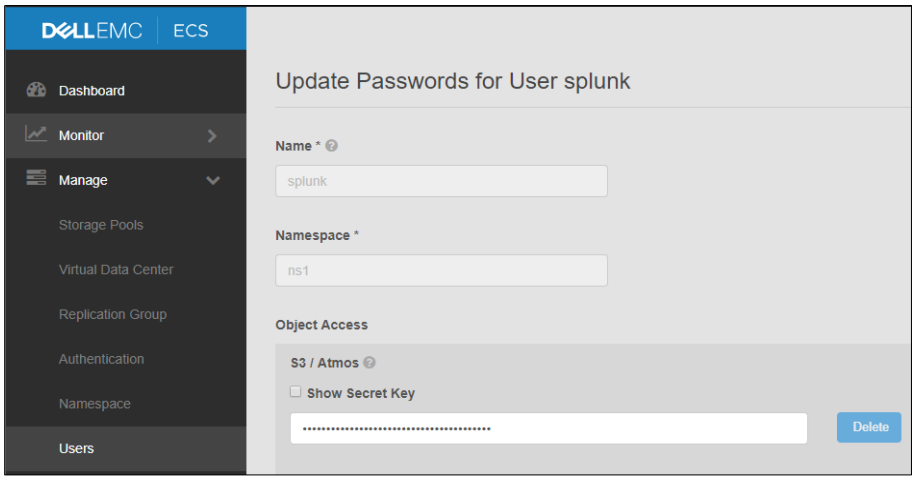

Figure 5 ECS Object User with an S3 secret key

2. Create an ECS bucket.

   a. Go to **Manage** > **Buckets** and select the namespace that you chose when creating the Object User (in our example that would be ns1). Click the **New Bucket** button.

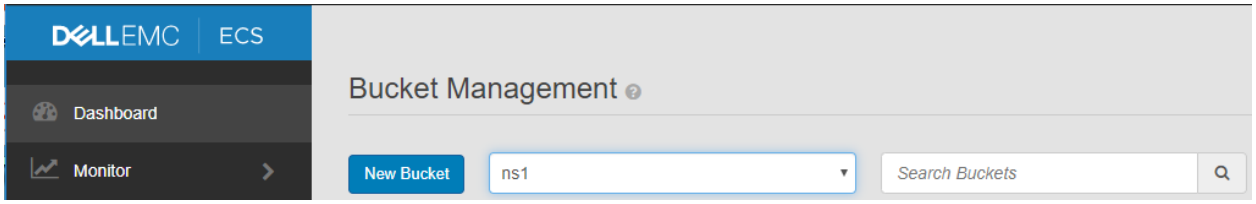

Figure 6 Create an ECS bucket.

   b. Enter a name and bucket owner. The bucket owner is the ECS Object user which was created in step 1.
   c. Click **Next** at the bottom of the page.

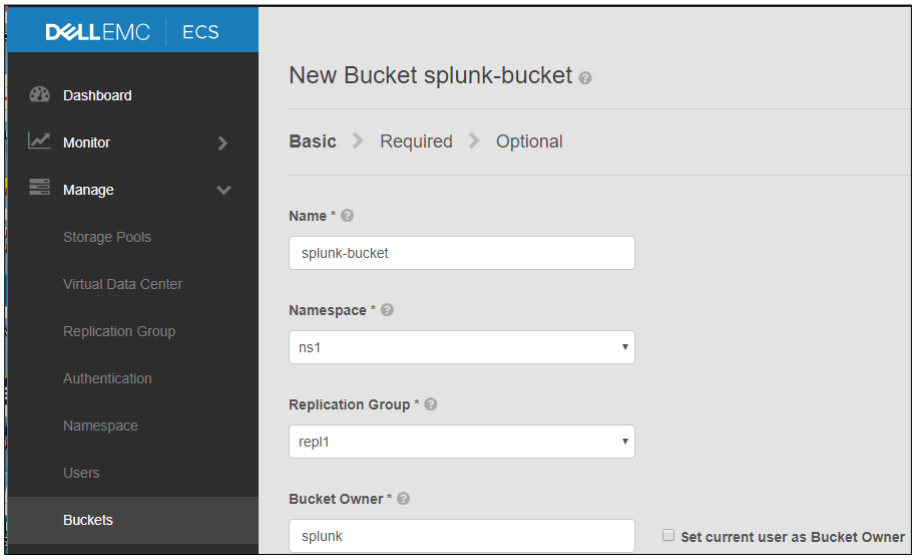

Figure 7 ECS Bucket name and owner

Note: Do not enable File System unless you have reviewed the best practices section of this document. The Required and Optional configuration settings can all be left as the defaults however the ECS documentation should be reviewed to better understand each option. Access During Outage can be enabled if the replication group chosen for the bucket spans multiple ECS sites. Reference the Temporary Site Outage section in the ECS Overview and Architecture document for details of this feature.

d. Click the **Save** button at the bottom of the final page to create the bucket.

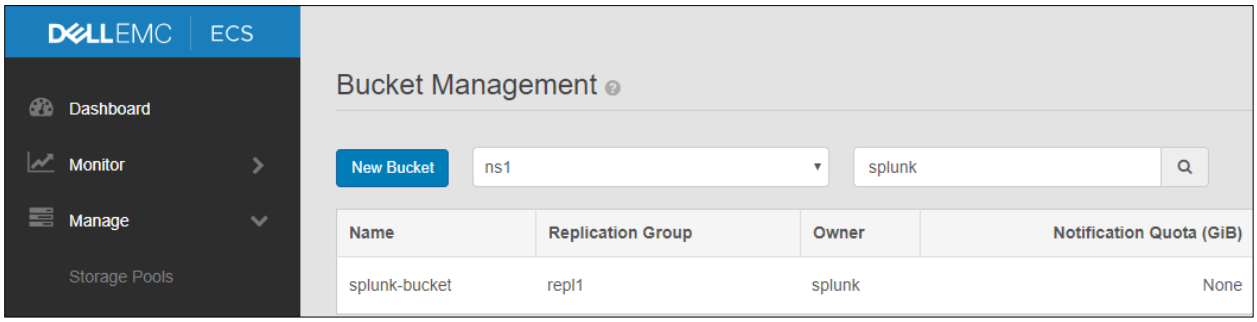

Figure 8 ECS bucket to store Splunk SmartStore warm index data

2.1.2 Configure SmartStore indexes with ECS

This section outlines the steps to configure a SmartStore index with ECS.

The SmartStore settings in indexes.conf enable and control SmartStore indexes. You can enable SmartStore for all an indexer's indexes, or you can enable it on an index-by-index basis, allowing a mix of SmartStore and non-SmartStore indexes on the same indexer.

Note: When you configure these settings on an indexer cluster's peer nodes, you must deploy the settings through the configuration bundle method. As with all settings in indexes.conf, SmartStore settings must be the same across all peer nodes.

This example configures SmartStore for an indexer cluster. On the master node, cd to /$SPLUNK_HOME/etc/master/_cluster/local and create a file named **indexes.conf**. The below SmartStore index example will store the Splunk index warm buckets (_audit, _internal, so on) on ECS.

Sample indexes.conf file:

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:ecs_store/$_index_name

repFactor = auto

# Configure the remote volume
[volume:ecs_store]
storageType = remote

# On the next line, the path attribute points to the remote storage location
# where indexes reside. Each SmartStore index resides directly below the
location
# specified by the path attribute. The <scheme> identifies a supported remote
# storage system type, such as S3. The <remote-location-specifier> is a
# string specific to the remote storage system that specifies the location
```

```
# of the indexes inside the remote system.
# This is an S3 example: "path = s3://mybucket/some/path".

path = s3://splunk-bucket/indexes

# The following S3 settings are required only if you're using the access and
secret
# keys. They are not needed if you are using AWS IAM roles.

remote.s3.access_key = splunk
remote.s3.secret_key = <ECS Object Users S3 Secret Key>
remote.s3.endpoint = <Endpoint to access ECS nodes>

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
thawedPath = $SPLUNK_DB/cs_index/thaweddb
# SmartStore-enabled indexes do not use coldPath, but you must still specify it
here.
coldPath = $SPLUNK_DB/cs_index/colddb
```

The highlighted parameters above are the required values that need to be modified to store index data in ECS.

Table 4 Indexes.conf configuration attributes

Attribute	Description
path	The above example is configured to use our bucket example in section 2.2.1, Step 2. A prefix or path named “indexes” is also being used. The bucket must be created on ECS prior to deploying the configuration bundle but any prefixes will automatically be created by SmartStore.
remote.s3.access_key	The ECS Object User name that was created in section 2.2.1
remote.s3.secret_key	The ECS Object Users S3 secret key generated in section 2.2.1
remote.s3.endpoint	The HTTP|HTTPS endpoint to access the ECS nodes. This would typically be the IP load balancer in front of the ECS cluster

Push the configuration bundle to one or more peer nodes from the master. This action may cause one or more peer nodes to restart.

2.2 Solution verification

There are several tools available which can be used to verify that the SmartStore indexes are being uploaded to ECS. In addition to the below methods, the Splunk logs contain insight into SmartStore operations.

Reference https://docs.splunk.com/Documentation/Splunk/7.2.6/Indexer/TroubleshootSmartStore for detailed troubleshooting methods.

2.2.1 Verify connectivity from SmartStore to ECS

SmartStore contains a CLI utility which can be used to verify connectivity to the remote store.

The syntax for running the command is:

```
./splunk cmd splunkd rfs -- ls --starts-with volume:ecs_store
```

The output of this command will list the contents of the ECS bucket.

2.2.2 Browse the ECS bucket using an S3 client.

S3 Browser is a freeware Windows client for S3 which can be used to browse ECS buckets.

To add a new account, select **Add new account** from the Accounts menu item. Provide an account name to identify the connection, select **S3 Compatible Storage,** and enter the endpoint to the ECS cluster. ECS supports both V2 and V4 Signatures. Lastly, enter you are ECS Object User ID and S3 password, ensure that **Use secure transfer (SSL or TLS)** is checked if your using HTTPS.

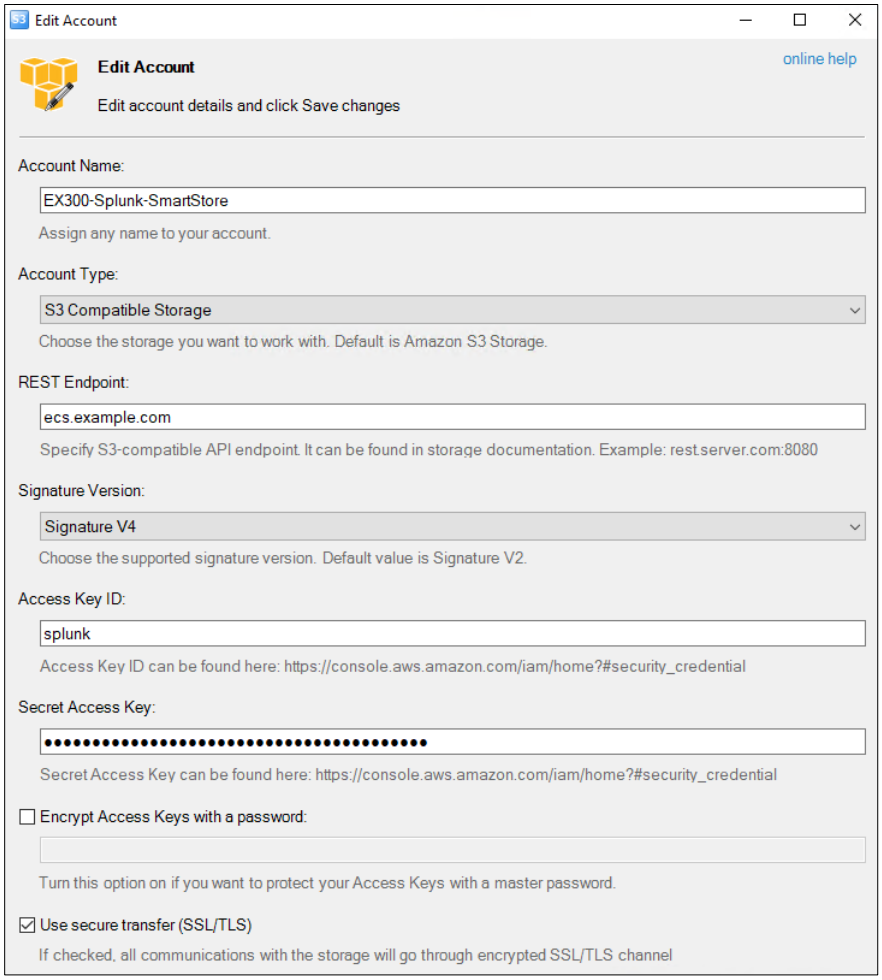

Figure 9 S3Browser configuration example

3 Best practices

The following best practices are recommended when configuring SmartStore indexes with Dell EMC ECS.

Recommendation	Details
Managing traffic flow to ECS	We recommend using an external IP traffic load balancer with ECS to manage the health and distribute traffic to all the ECS nodes in the cluster.
remote.s3.use_delimiter to improve listing performance	ECS Release 3.2.2 is the first release that works with the default value for the SmartStore parameter **remote.s3.use_delimiter**. This means that ECS Release 3.2.2 or greater cannot be used with file system enabled buckets when this parameter is enabled, as file system semantics for ECS native NFS and Hadoop support require a standard “/” delimiter in file paths. Disabling this parameter for use with legacy versions of ECS or file system enabled buckets has not been tested or is it a best practice for ECS performance purposes.
Local storage requirements	Splunk recommends to, at a minimum, provision enough storage to keep at least 7 to 10 days of data in cache, as searches typically occur on data indexed within the last 7 to 10 days. Review the local storage requirements section in the Splunk 7.3 documentation for guidance
Cold Buckets	SmartStore index buckets generally roll to frozen directly from warm. Cold buckets can exist in a SmartStore-enabled index, but only under limited circumstances. If you migrate an index from non-SmartStore to SmartStore, any migrated cold buckets use the existing cold path as their cache location, postmigration. In all respects, cold buckets are functionally equivalent to warm buckets. The cache manager manages the migrated cold buckets in the same way that it manages warm buckets. The only difference is that the cold buckets will be fetched into the cold path location, rather than the home path location.
Data Retention	Data retention policy for SmartStore indexes on indexer clusters is configured using settings similar to those for non-SmartStore indexes. However, with SmartStore indexes, data retention is managed cluster-wide, rather than on a per-indexer basis.
Multipart upload or download part size	The default settings for multipart download and upload part size are 128 MB and should not be modified unless that value has proven to improve throughput.
Versioning	SmartStore supports versioning, used primarily for frozen data, that exceeds the configured data retention time. When the parameter **remote.s3.supports_versioning=false**, SmartStore will put a delete marker on any corresponding bucket that gets frozen and this data or bucket is ignored by SmartStore for any subsequent searches. This prevents any accidental deletion. **Note:** The ECS bucket being used to store the index data must have versioning enabled.

Recommendation	Details
	N**ote:** It is recommended to set a life-cycle policy on the ECS bucket to eventually remove the SmartStore frozen bucket and free capacity. If **remote.s3.supports_versioning=true**, which is the default, then the data is deleted by SmartStore once the data ages out and is frozen.

A Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell EMC storage platforms.

A.1 Related resources

ECS product documentation

- Dell EMC ECS product documentation
  - https://community.emc.com/docs/DOC-73931
- Dell EMC ECS Architecture and Overview
  - http://www.emc.com/collateral/white-papers/h14071-ecs-architectural-guide-wp.pdf
- Dell EMC ECS Networking and Best Practices
  - http://www.emc.com/collateral/white-paper/h15718-ecs-networking-bp-wp.pdf
- Dell EMC ECS Best Practices
  - https://www.emc.com/collateral/white-papers/h16016-ecs-best-practices-guide-wp.pdf

Splunk SmartStore documentation

- Splunk SmartStore Deployment and Configuration Guide
  - https://docs.splunk.com/Documentation/Splunk/7.3.0/Indexer/AboutSmartStore
- SmartStore Architectural Overview
  - https://docs.splunk.com/Documentation/Splunk/7.3.0/Indexer/SmartStorearchitecture
- Restrictions when using SmartStore indexes
  - https://docs.splunk.com/Documentation/Splunk/7.3.0/Indexer/ConfigureSmartStore#Settings_in_indexes.conf_that_are_incompatible_with_SmartStore_or_otherwise_restricted
- SmartStore Troubleshooting Guide
  - https://docs.splunk.com/Documentation/Splunk/7.3.0/Indexer/TroubleshootSmartStore
- Configuring Data Retention for SmartStore indexes
  - https://docs.splunk.com/Documentation/Splunk/7.3.0/Indexer/SmartStoredataretention