Abstract
This document describes how to back up and restore Elasticsearch® data to Dell EMC™ ECS storage.

July 2019
Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2019</td>
<td>Initial release</td>
</tr>
</tbody>
</table>

Acknowledgements

This paper was produced by the Unstructured Technical Marketing Engineering and Solution Architects team.

Author: Rich Paulson

The information in this publication is provided “as is.” Dell Inc. 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 software described in this publication requires an applicable software license.

Copyright © 2019 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. [7/12/2019] [Configuration and Deployment] [H17847]
# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revisions</td>
<td>2</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>2</td>
</tr>
<tr>
<td>Table of contents</td>
<td>3</td>
</tr>
<tr>
<td>Executive summary</td>
<td>4</td>
</tr>
<tr>
<td>Objectives</td>
<td>4</td>
</tr>
<tr>
<td>Audience</td>
<td>4</td>
</tr>
<tr>
<td>1 Solution overview</td>
<td>5</td>
</tr>
<tr>
<td>1.1 Solution architecture</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Key components</td>
<td>5</td>
</tr>
<tr>
<td>2 Solution implementation</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Implementation workflow</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Installation and configuration steps</td>
<td>6</td>
</tr>
<tr>
<td>2.2.1 Step 1: Create a bucket in ECS to store the snapshot data</td>
<td>6</td>
</tr>
<tr>
<td>2.2.2 Step 2: Install the S3 plugin</td>
<td>6</td>
</tr>
<tr>
<td>2.2.3 Step 3: Register ECS as a repository</td>
<td>7</td>
</tr>
<tr>
<td>2.2.4 Step 4: Take a snapshot (backup) of an Index</td>
<td>8</td>
</tr>
<tr>
<td>2.2.5 Step 5: Restore a snapshot to the same cluster</td>
<td>8</td>
</tr>
<tr>
<td>3 Configuration and tuning</td>
<td>10</td>
</tr>
<tr>
<td>A Technical support and resources</td>
<td>11</td>
</tr>
<tr>
<td>A.1 Related resources</td>
<td>11</td>
</tr>
</tbody>
</table>
Executive summary

The explosive growth of unstructured data and cloud-native applications has created demand for scalable cloud storage infrastructure in the modern datacenter. Dell EMC™ ECS is the third-generation object store platform from Dell EMC. ECS is designed from the ground up to deliver modern cloud storage API, distributed data protection, and active/active availability spanning multiple data centers.

Elasticsearch® is a distributed, RESTful search and analytics engine capable of storing your data, and includes a smart solution to back up single indices or entire clusters to a remote shared filesystem, S3 or HDFS.

Objectives

This document illustrates how configure Elasticsearch to store data in ECS using the Elasticsearch backup and restore API.

Audience

This document is intended for administrators who manage Elasticsearch deployments. This guide assumes a high level of technical knowledge for the devices and technologies described.
1 **Solution overview**

This section provides an overview of the Dell EMC ECS integration with Elasticsearch including the key technologies used.

1.1 **Solution architecture**

The below figure illustrates the architectural workflow used in this guide.

![Architecture Workflow](image)

1.2 **Key components**

The following components and versions were used for the examples in this guide.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Dell EMc components</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Component</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>ECS appliance</td>
<td>EX300 (5 Nodes)</td>
</tr>
<tr>
<td>ECS version</td>
<td>3.3.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Elasticsearch components</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Component</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Elasticsearch</td>
<td>7.2.0</td>
</tr>
<tr>
<td>Kibana</td>
<td>7.2.0</td>
</tr>
<tr>
<td>Logstash</td>
<td>7.2.0</td>
</tr>
</tbody>
</table>
2 Solution implementation

This section describes the high-level steps required to configure Elasticsearch to store snapshot data to ECS. That this guide assumes Elasticsearch and Kibana (optional) have been installed and are functioning.

That the ELK stack was installed and configured to ingest ECS access logs using the document Insights into ECS Data Utilization Using Open Source Tools.

2.1 Implementation workflow

The below figure outlines the steps to implement this solution.

- Step 1: Create the ECS bucket to store the snapshot data
- Step 2: Install the Elasticsearch S3 repository plugin
- Step 3: Register ECS as a repository
- Step 4: Take a snapshot of an index
- Step 5: Restore a snapshot to the same cluster

Figure 2 Implementation steps

2.2 Installation and configuration steps

The following steps are meant to be a guide for configuring ECS as an Elasticsearch S3 repository.

2.2.1 Step 1: Create a bucket in ECS to store the snapshot data

The first step is to create a bucket to store the snapshots. This can be done using the ECS Web Portal, ECS API or an S3 client such as S3 Browser. Reference the ECS Administration Guide for detailed information on creating buckets.

2.2.2 Step 2: Install the S3 plugin

The S3 repository plugin adds support to use ECS as the repository for Snapshots and Restores. Reference the Elasticsearch S3 Repository Plugin documentation for instructions on installing the plugin.

To verify that the plugin has been installed invoked the below API call:
```
[root@elk ~]# curl -X GET "localhost:9200/_cat/plugins?v8s=component&h=name,component,version,description"
[  "name": "component",  "version": "9.7.0",  "description": "The S3 repository plugin adds S3 repositories"
```

Figure 3 Elasticsearch S3 plugin
2.2.3 Step 3: Register ECS as a repository

Elasticsearch provides REST APIs to register an S3 repository. The below example registers ECS as a repository using the _snapshot API.

**Note:** By default, S3 repositories use a client named default, but this can be modified using the repository setting client. For example: client: "my_ecs_client".

Create the S3 repository

```bash
curl -X PUT "localhost:9200/_snapshot/ecs" -H 'Content-Type: application/json' -d '
{
  "type": "s3",
  "settings": {
    "bucket": "<ECS Bucket>",
    "base_path": "snapshots",
    "endpoint": "<ECS Endpoint>",
    "access_key": "<ECS Object User>",
    "secret_key": "<ECS S3 Secret Key>",
    "protocol": "http"
  }
}
'
```

The following examines above settings in more detail:

- **bucket**: This the name of the bucket that was created in Step 1 to store snapshot data.
- **base_path**: [optional] This specifies the path within the bucket to store data.
- **endpoint**: A load balancer is required with ECS. This specifies the endpoint of the load balancer in front of the ECS nodes.
- **access_key**: Specifies the ECS object user to use for authentication.
- **secret_key**: The secret s3 key of the ECS object user.
- **protocol**: Specifies whether HTTP or HTTPS is used.

**Note:** Large files can be broken down into chunks if needed. The default setting is 1gb which should not need to be adjusted for ECS.

Additional settings can be specified for the repository and are specified in the Elasticsearch [Repository settings](https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-repository.html) documentation.

We can verify that our repository was successfully created by using the ‘POST /_snapshot/ecs/_verify’ call.

```
[root@elk ~]# curl -XPOST 'http://localhost:9200/_snapshot/ecs/_verify?format=json&pretty'
{
  "nodes": {
    "y80Xw2nZTcm4_cVREGoQA": {
      "name": "node-1"
    }
  }
}
```

**Figure 4** S3 Repository verification
2.2.4 Step 4: Take a snapshot (backup) of an Index

The next step is to take a snapshot of the logstash index which contains the ECS data access log events. Our index is named logstash-2019.07.01-000001.

Index snapshot

curl -X PUT "localhost:9200/_snapshot/ecs/snapshot_1" -H 'Content-Type: application/json' -d
{
  "indices": "logstash-2019.07.01-000001",
  "ignore_unavailable": true,
  "include_global_state": false
}

The status of the snapshot can be verified using the below API.

We can verify that snapshot data exists in the ECS bucket using an S3 client such as s3cmd, s3curl or S3 Browser. The below show the properties of our bucket storing the snapshot data.

Figure 5   Snapshot data

2.2.5 Step 5: Restore a snapshot to the same cluster

A snapshot can be restored using the following command:

POST /_snapshot/ecs/snapshot_x/_restore

This example restores one of our snapshots to the same cluster. Since an existing index can be only restored if it is closed and has the same number of shards as the index in the snapshot we’ll restore our snapshot to a new index.

Note: Reference the Elasticsearch Snapshot and Restore documentation for detailed information on restoring indices.
List the snapshots in the ECS repository using the below API.

```
root@elk -> $ curl -X GET "localhost:9200/_cat/snaps?v&v=s=1d"
```

The API call used to restore snapshot_1 to a new index is as follows:

```
curl -X POST "localhost:9200/_snapshot/ecs/snapshot_1/_restore" -H 'Content-Type: application/json' -d '{
    "indices": "logstash-2019.07.01-000001",
    "ignore_unavailable": true,
    "include_global_state": true,
    "include_aliases": false,
    "rename_pattern": "logstash(.+)",
    "rename_replacement": "restored_index_$1"
}
```

Verify that the snapshot was restored to a new index:

```
root@elk -> $ curl -X GET "localhost:9200/_cat/indices?v"
```

**Note:** Our index health is yellow because the current cluster only consists of a single node, so the replicas remain unassigned simply because no other node is available to contain them.
### Configuration and tuning

The following configuration and tuning changes are recommended.

<table>
<thead>
<tr>
<th>Description</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3 repository chunk size</td>
<td>Big files can be broken down into chunks during snapshotting if needed. The chunk size can be specified in bytes or by using size value notation, i.e. 1gb, 10mb, 5kb. Defaults to 1gb. Dell EMC recommends using the default setting of 1gb.</td>
</tr>
<tr>
<td>S3 repository buffer size</td>
<td>The Minimum threshold below which the chunk is uploaded using a single request. Beyond this threshold, the S3 repository will use Multipart upload to split the chunk into several parts, each of buffer_size length, and to upload each part in its own request. Dell EMC recommends using the minimum default setting between 100mb and 5% of the heap size.</td>
</tr>
</tbody>
</table>
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
- Dell EMC ECS Networking and Best Practices
- Dell EMC ECS Best Practices
- Insights into ECS Data Utilization using Open Source Tools

Elasticsearch product documentation

- Introduction to Elasticsearch
- Elasticsearch S3 plugin
- Elasticsearch S3 Repository Settings