File Access Manager Data Backup

Version: 8.4

Revised: March 27, 2023

Copyright and Trademark Notices

Contents

Elasticsearch Backup Overview

There are two types of Elasticsearch repositories and both are the File System type. For more information, read https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshots-filesystem-repository.html.

The two repositories are:

Continuous_backup

Used for backing up the whole cluster. This repository holds snapshots that are being taken every hour with the following name format "fam-backup-yyyy.MM.dd-hh:mm:ss-UUID." Every snapshot will be saved for 60 days.

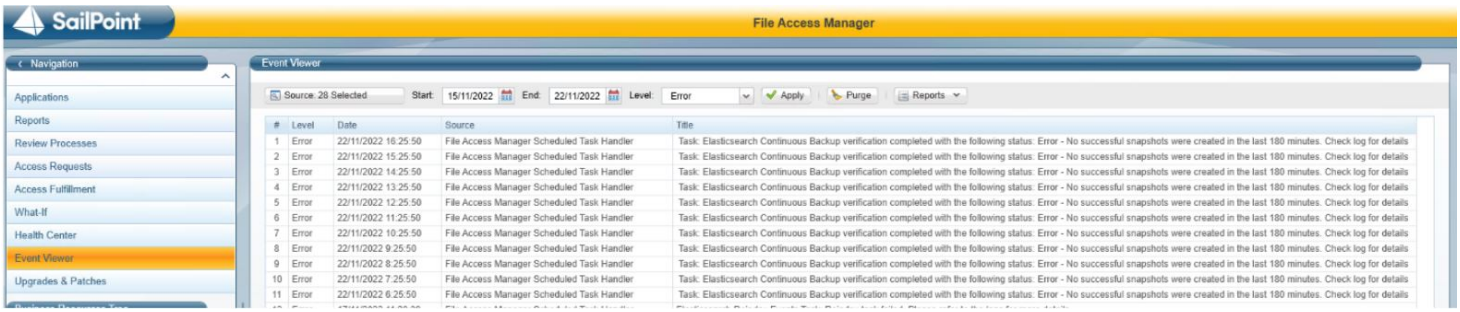
This repository can contain up to 1500 snapshots (in case of also creating snapshots manually) and minimum of 100.

Retention_backup

Used for backing up the events indices which are deleted in the activity data retention process. A snapshot of the deleted indices will be created with the following name format: “retention_backup-yyyy.MM.dd-hh:mm:ss." These snapshots will be saved forever.

Continuous Backup Monitoring

The continuous backup repository monitoring is run by the File Access Manager Scheduled Task Handler service. The monitoring starts 3 hours after the service starts and monitors the continuous_backup snapshots once an hour. The results are displayed in the File Access Manager Admin Client Event Viewer. In case of errors, more detailed results can be found in the File Access Manager Scheduled Task Handler log and the Elasticsearch log.



There is also an option to generate a report which contains more details in case of failures, including the last successful snapshot name.

Elasticsearch Continuous Backup Monitoring Configurations

> The monitoring configurations should not be changed.

In case there is a temporary need to change them, it can be done in the TaskScheduler's App.cofig file:

- elasticBackupHealthMonitoringPoolingIntervalInSeconds – the interval to check the "continuous_backup" snapshots (default is 3600 seconds – 1 hour)
- elasticBackupHealthMonitoringDueTimeInSeconds – the due time before starting to monitor the "continuous_backup" snapshots (default is 7200 seconds – 2 hours. Snapshots will start being taken one hour after enabling the backup)
- maxTimeSinceLastSuccessfulSnapshotInMinutes – the period since the last successful snapshots before the backup status will be set to Failure (default is 180 minuets – 3 hours)

Note: Useful command for manual monitoring is ‘GET _slm/policy/fam-backup’
Example of output is:


```
{
  "fam-backup": {
    "version":1,
    "modified_date_millis":1668585570288,
    "policy": {
      "name":"<fam-backup-{now{yyyy.MM.dd-hh:mm:ss}}>",
      "schedule":"0 0 * * * ?",
      "repository":"continuous_backup",
      "config": {
        "indices":"_all"
      },
      "retention": {
        "expire_after":"60d",
        "min_count":100,
        "max_count":1500
      }
    },
    "last_success": {
      "snapshot_name":"fam-backup-2022.11.25-12:59:59--ae1qogiqmiiz24dgzflvq",
      "start_time":1669381199972,
      "time":1669381203577
    },
    "last_failure": {
      "snapshot_name":"fam-backup-2022.11.27-07:59:59-o3u5etmrtsgrakbvkek9vq",
      "time":1669536000677,
      "details":"{\"type\":\"snapshot_exception\",\"reason\":\"[continuous_backup:fam-backup-2022.11.27-
      07:59:59-o3u5etmrtsgrakbvkek9vq/D6CAvlB9SmmNpqW0GiqAMw] failed to update snapshot in
      repository\",\"caused_by\":{\"type\":\"file_system_exception\",\"reason\":\"\\\\\\\\\FAM-
      ANATOLY7\\\\repo\\\\Prod\\\\t14\\\\continuous_backup\\\\snap-D6CAvlB9SmmNpqW0GiqAMw.dat: There is not
      enough space on the disk\",\"suppressed\":
      [{\"type\":\"file_system_exception\",\"reason\":\"\\\\\\\\\FAM-
      ANATOLY7\\\\repo\\\\Prod\\\\t14\\\\continuous_backup\\\\meta-D6CAvlB9SmmNpqW0GiqAMw.dat: There is not
      enough space on the disk\"}]}}"
    },
    "next_execution_millis":1669539600000,
    "stats": {
      "policy":"fam-backup",
      "snapshots_taken":130,
      "snapshots_failed":119,
      "snapshots_deleted":0,
      "snapshot_deletion_failures":0
    }
  }
}
```

Elasticsearch Backup Installation

> Important: Before you install the Elasticsearch backup, make sure that Elasticsearch has access and proper permissions to the backup folder.

> Note: Some setting are applied through a task outside the Server Installer which could result in longer completion time. Verify all changes were applied.

> Note: It is recommended not trying to change multiple settings at the same time. Change each setting separately and verify it was completed successfully before continuing with the next change.

To set permissions to the backup folder:

1. Set the folder as share:

   Folder Properties > Sharing > Share... > Share.

2. Share the backup base path with all master and data nodes:

   Folder Properties > Sharing > Advanced Sharing...

3. Check **Share this folder**.

4. Navigate to **Permissions > Add... > Object Types...**.

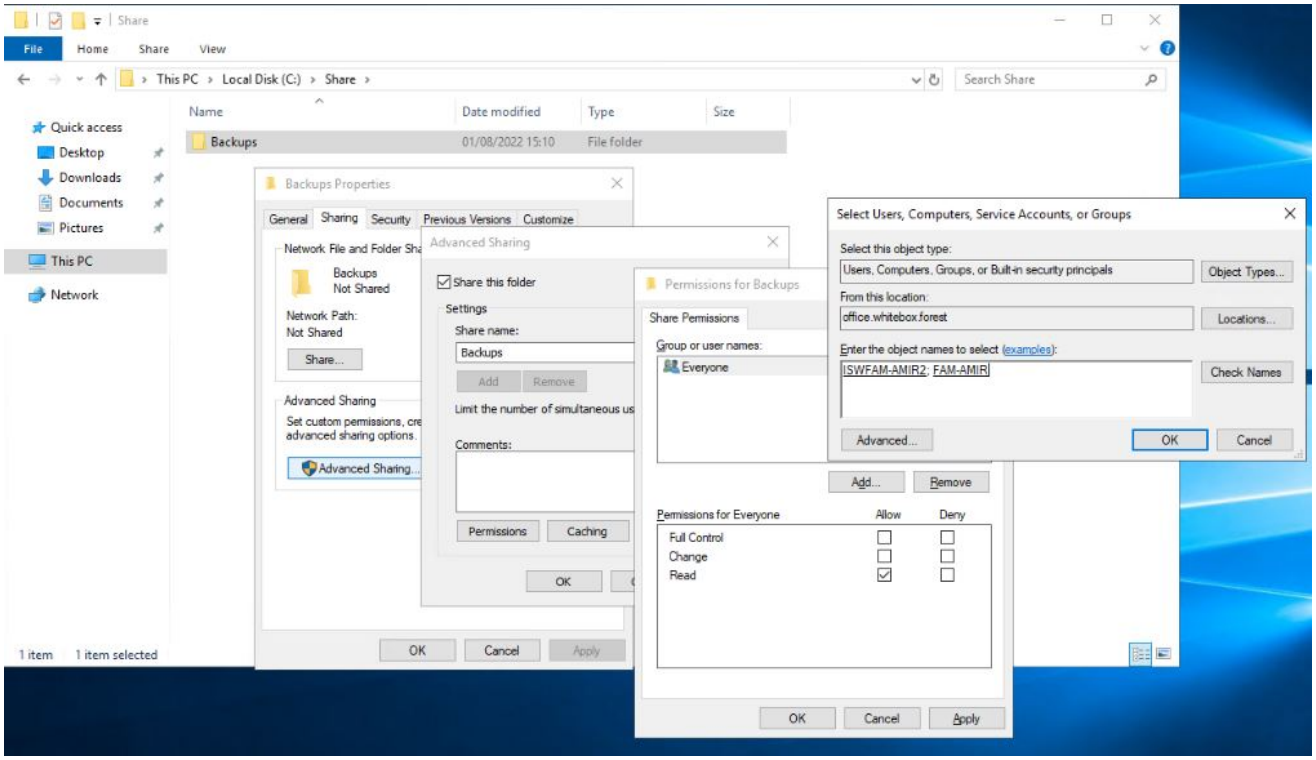
5. Check **Computers**.

6. Select **OK**.

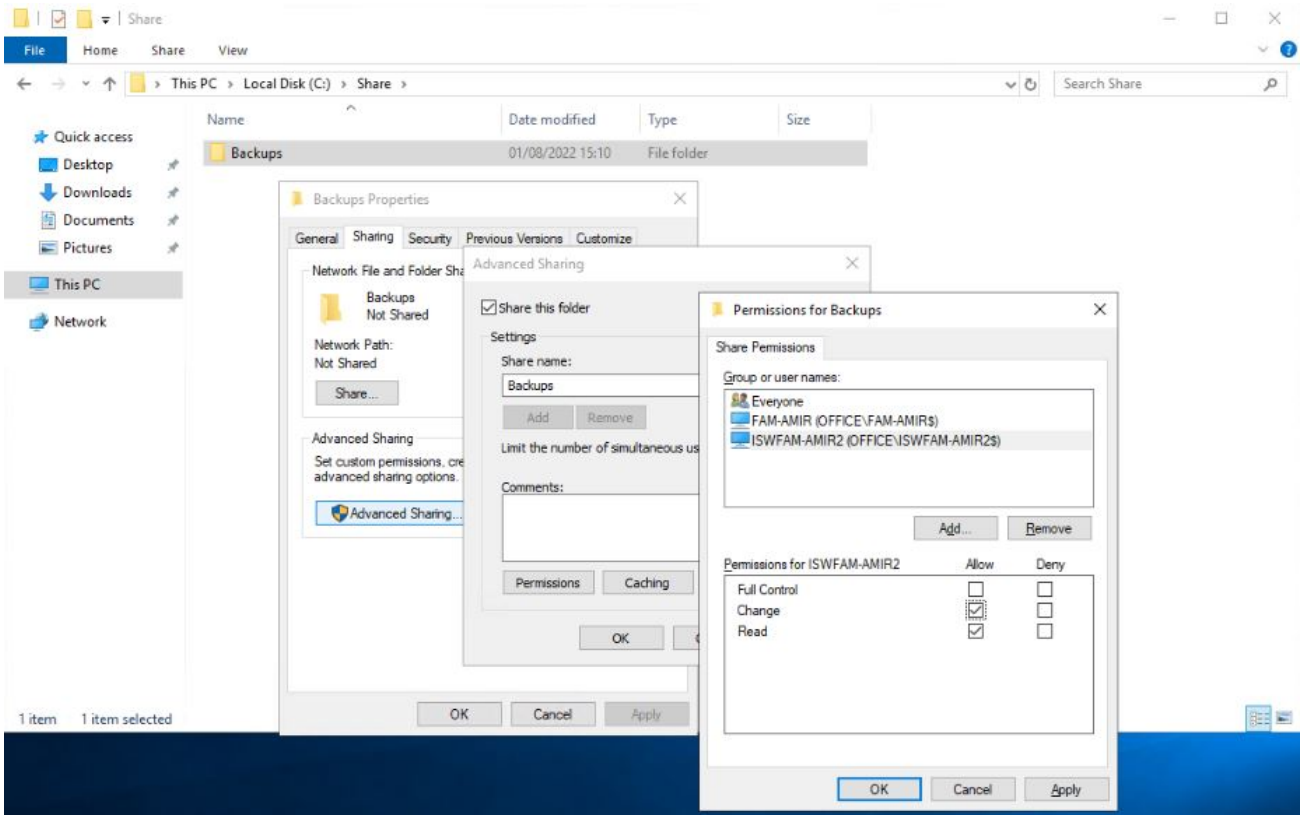
7. Add all the master and data nodes (MACHINE_NAME$).

   > Note: It is recommended to have an odd number of nodes. Ideally 3 and above.

8. Select **OK**.

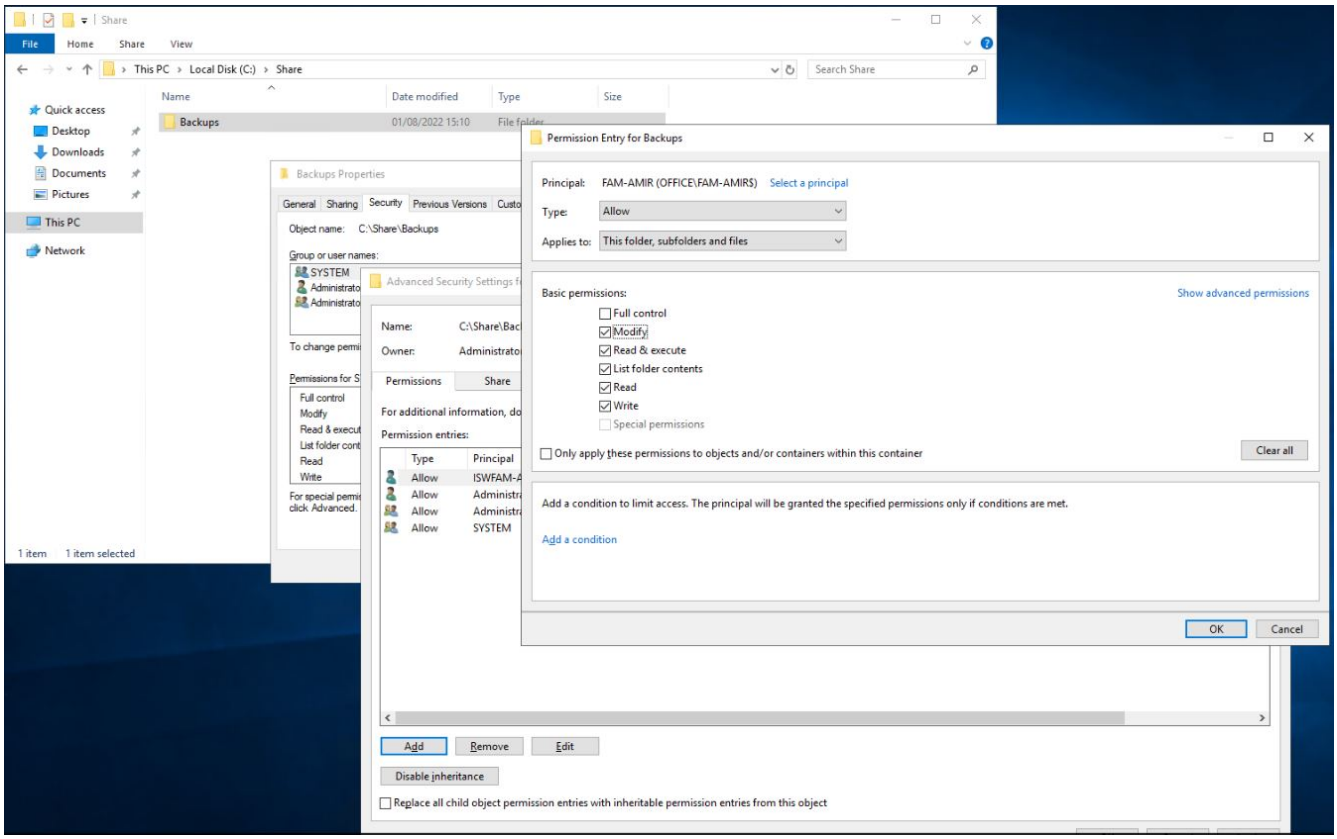


9. In the Permissions window, give each node a **Change** permission.



10. Select **OK > OK > Close**.
11. Set the shared folder's (e.g., "Backups") NTFS security settings to **Modify** for all master and data nodes. Do this by navigating to **Folder Properties > Security > Advanced > Permissions tab > Add** .
12. Select a principal.
13. Navigate to **Object Types**.
14. Select **Computers** and then select **OK**.
15. Add one of the nodes (MACHINE_NAME$) and select **OK**.
16. Set Basic permissions to Modify and select **OK**.

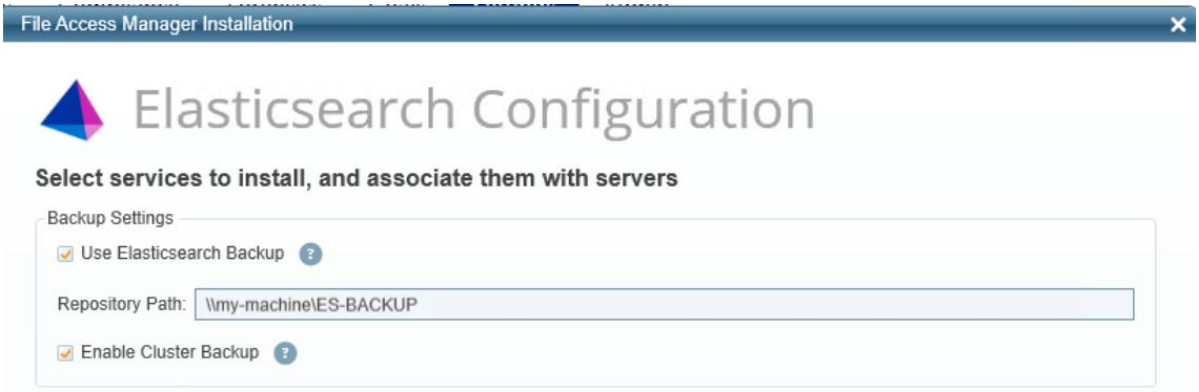
> Important: Do this for all master and data nodes.



17. If the Elasticsearch and the backup folder are in different domains, the domains should have trust between them.

Backup Elasticsearch Configuration

1. In the Elasticsearch Configuration window of the Server Installer, select the desired configuration.



2. Select the **Use Elasticsearch Backup** to enable backup of elastic data. If this option is unchecked, there will be no kind of backup of elastic data. This setting is also applied on the Elasticsearch disaster recovery backup. Without the backup, there will be no disaster recovery.

3. If Use Elasticsearch Backup is selected, you must provide the Repository Path. This folder should be configured and have the appropriate permissions according to the above section.

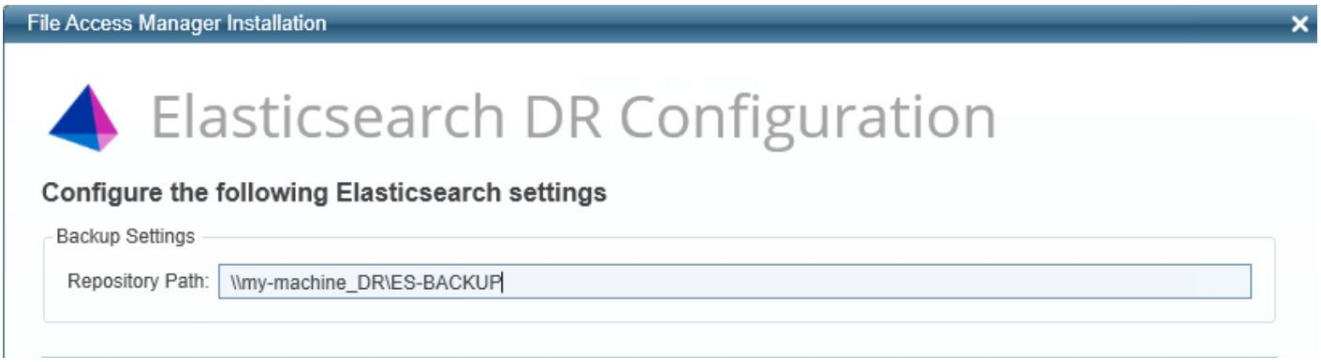
   Any change in the repository path will cause a restart of each elastic node.

4. Select **Enable Cluster Backup** to run the continuous_backup and take snapshots of the full cluster. This setting is also applied on the Elasticsearch disaster recovery backup.

5. If a disaster recovery environment is configured and if Use Elasticsearch Backup was checked, you must provide the Repository Path for the disaster recovery Elasticsearch.

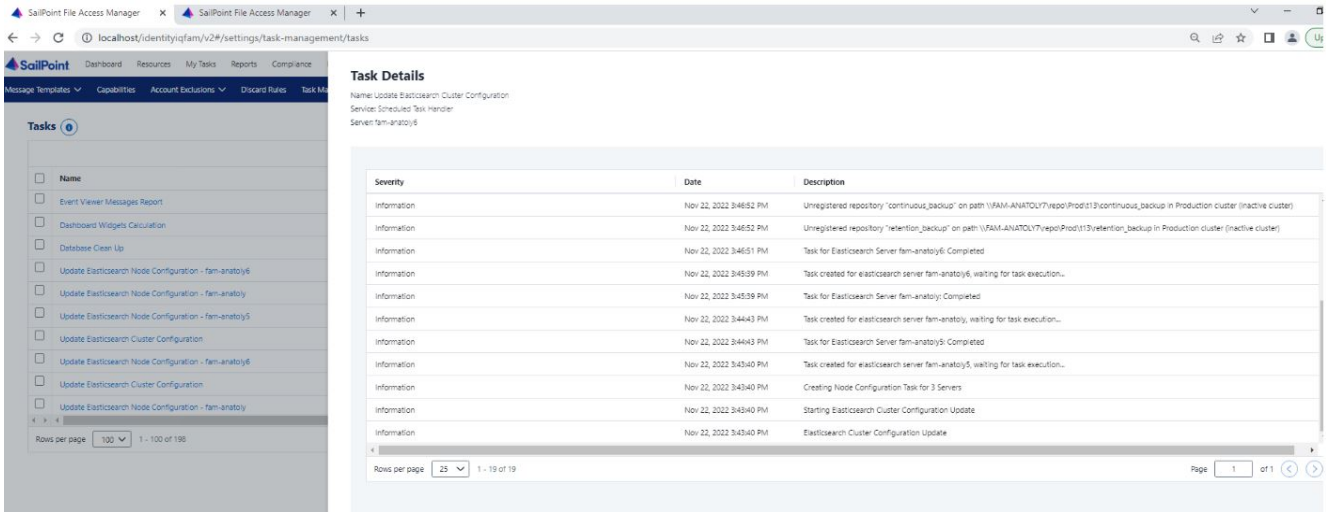
   This folder should be configured and have the appropriate permissions according to the above section.

> Note: The disaster recovery repository path must be different then the production repository path.



If any change occurs in the backup configuration, it is highly recommended to keep track of the continuous backup monitoring in the File Access Manager Admin client event viewer. Make sure nothing went wrong and that snapshots are being created successfully.

Elasticsearch backup configuration changes are performed in an asynchronous way by a task named Update Elasticsearch Cluster Configuration by the Scheduled Task Handler Service.



Data Restoration

More information about restoring Elasticsearch data can be found here.

Considerations

Keep the following in mind when restoring data from a snapshot:

- You can only restore an existing index if it's closed and the index in the snapshot has the same number of primary shards
- You cannot restore an existing open index
- The restore operation automatically opens restored indices

1. Get a list of available snapshots ordered by descending start time.
    a. GET _snapshot/continuous_backup/*?order=desc
    b. GET _snapshot/retention_backup/*?order=desc
2. Get a list of available snapshots from a specific date
    a. GET _snapshot/continuous_backup/fam-backup-2022.08.02-*?verbose=false
    b. GET _snapshot/retention_backup/retention_backup-2022.08.02-*?verbose=false

Restore a Deleted Index

To restore a deleted index or indices, find the specific snapshots which contain the index you want to restore (you can use the above examples to find the relevant snapshot).

```
POST _snapshot/retention_backup/retention-backup-2022.08.01-00:10:00/_restore
{
 "indices": "events_2022_07_2, events_2022_05_1"
}
```

Restore an Existing Index

If needing to restore an existing index, there are two preferable ways to do it:

1. Delete and Restore

   https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshots-restore-snapshot.html#delete-restore

   In case you only need to restore a specific index, the simplest way to avoid conflicts is to delete an existing index before restoring it.

   Example: DELETE pii-1, pii-8

   In the restore request, explicitly specify the repository name, snapshot name, and any indices to restore.

```
POST _snapshot/continuous_backup/fam-backup-2022.08.03-09:00:00-
fv59i0lpqjipxdtcwirs8a/_restore
{
 "indices": "pii-1", "pii-8"
 }
```

2. Rename and Restore

   https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshots-restore-snapshot.html#rename-on-restore

   If you want to avoid deleting existing data, you can instead rename the indices you restore. You typically use this method to compare existing data to historical data from a snapshot. For example, you can use this method to review documents after an accidental update or deletion.

```
POST _snapshot/my_repository/my_snapshot_2099.05.06/_restore
{
 "indices": "my-index,logs-my_app-default",
 "rename_pattern": "(.+)",
 "rename_replacement": "restored-$1"
 }
```

   When the restore operation is complete, you can compare the original and restored data. If you no longer need an original index, you can delete it and use a reindex to rename the restored one.

   a. To delete the original index: DELETE my-index

b. To reindex the restored index and rename it: POST _reindex

```
{
 "source": {
   "index": "restored-my-index"
  },
  "dest": {
   "index": "my-index"
  }
}
```

Restore an Entire Cluster

> Caution: This should only be used in case of a failure.

> Note: File Access Manager recommends reading the Elasticsearch guide first which can be accessed here.

1. Temporarily stop indexing and turn off the following features:

GeoIP database downloader

```
PUT _cluster/settings
{
 "persistent": {
   "ingest.geoip.downloader.enabled": false
 }
}
```

ILM

```
POST _ilm/stop
```

Monitoring

```
PUT _cluster/settings
{
 "persistent": {
  "xpack.monitoring.collection.enabled": false
 }
}
```

Machine Learning

```
POST _ml/set_upgrade_mode?enabled=true
```

Watcher

```
POST _watcher/_stop
```

2. Use the cluster update settings API to set **action.destructive_requires_name to false**. This allows you delete data streams and indices using wildcards.

```
PUT _cluster/settings
{
 "persistent": {
  "action.destructive_requires_name": false
 }
}
```

3. Delete all existing data streams on the cluster.

```
DELETE _data_stream/*?expand_wildcards=all
```

4. Delete all existing indices on the cluster.

```
DELETE *?expand_wildcards=all
```

5. Restore the entire snapshot, including the cluster state. By default, restoring the cluster state also restores any feature states in the snapshot.

```
POST _snapshot/my_repository/my_snapshot_2099.05.06/_restore
{
 "indices": "*",
 "include_global_state": true
}
```

> Note: Restore request return immediately. The restore happens in the background and the user needs to wait while it completes.

The GET _cluster/health request can be used to monitor Cluster Health and restore progress. See below for the Health request example of response. Green status indicates that the cluster is fine and the restore is complete.

6. When the restore operation is complete, resume indexing and restart any features you stopped.

GeoIP database downloader

```
PUT _cluster/settings
{
 "persistent": {
  "ingest.geoip.downloader.enabled": true
 }
}
```

ILM

```
POST _ilm/start
```

Machine Learning

```
POST _ml/set_upgrade_mode?enabled=false
```

Monitoring

```
PUT _cluster/settings
{
 "persistent": {
  "xpack.monitoring.collection.enabled": true
 }
}
```

Watcher

```
POST _watcher/_start
```

7. Reset the action.destructive_requires_name cluster setting.

```
PUT _cluster/settings
{
 "persistent": {
  "action.destructive_requires_name": null
 }
}
```

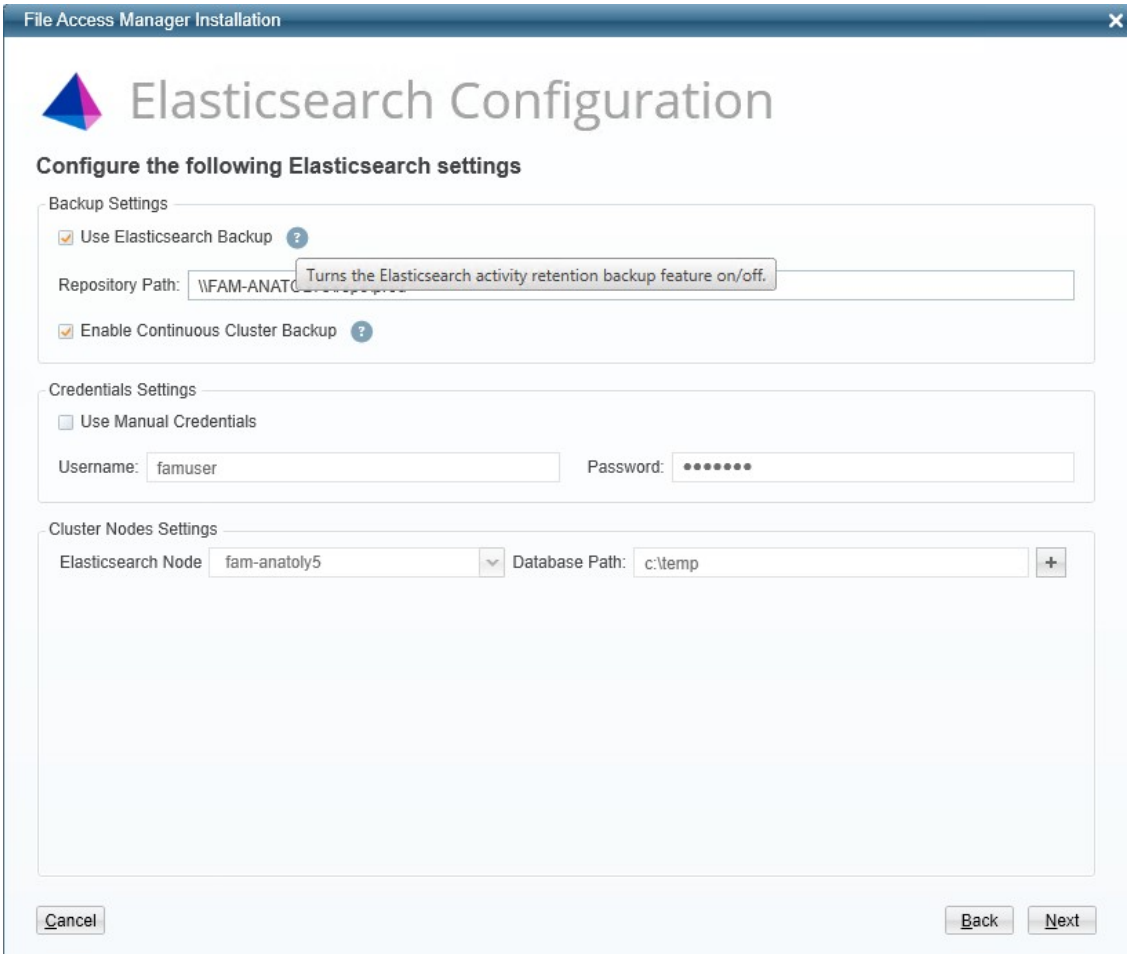
Retention Backup

In case backup was enabled in the system level (see Backup Elasticsearch Configuration), snapshots of activities indices will be taken for any of application which have Activity Data Retention configured.

The snapshot will contain all the indices that should be deleted and will be created before the retention deletion process takes place. The snapshot repository and format is in Elasticsearch Backup Overview.

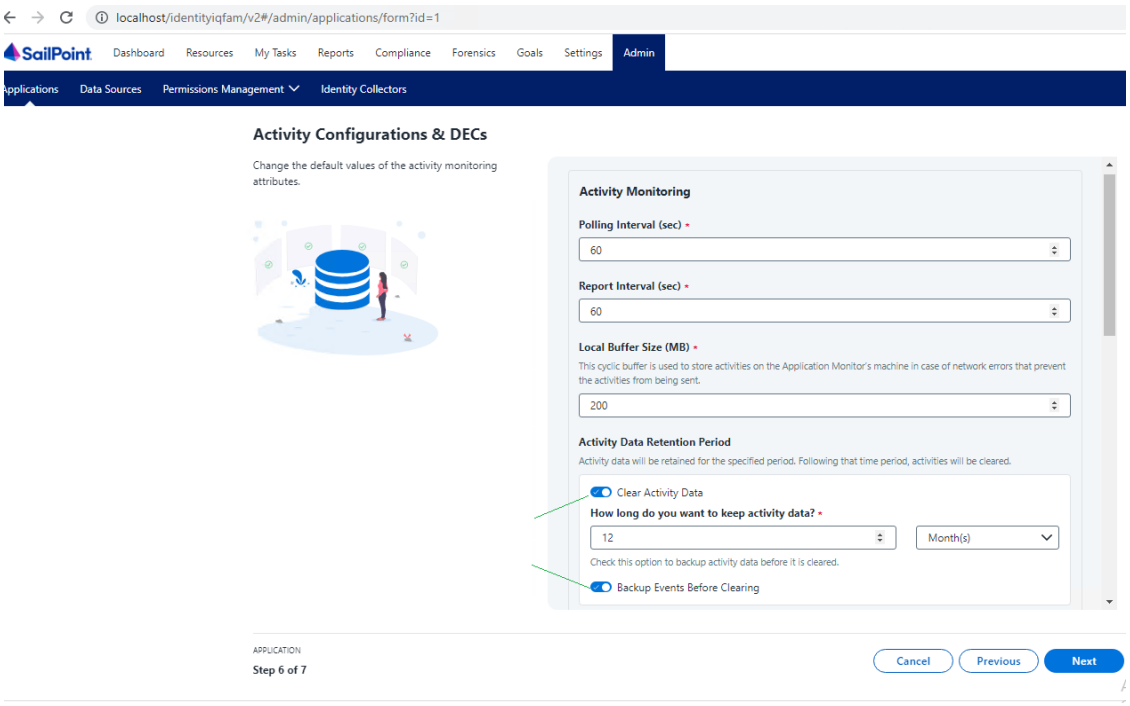
To enable retention backup, the following steps should be performed:

1. In the Server Installer, select the **Use Elasticsearch Backup option**.



2. When setting up an application, on the Activity Configurations and DECs screen, enable **Clear Activity Data** and Backup **Events Before Clearing**.

   Provide the time frame for keeping the backup data.



3. Within the Tasks screen, initiate the Activity Data Retention Cleanup task.

   Running this task will:

   - Mark indices for deletion for each configured application
   - Back indices to the "retention_backup" folder

- Delete the backup indices