FeatureStore-CompactUtil-Solution

Introduction:

The compaction utility is a python script that takes in a feature group name and compacts the offline store for that feature group. This script is run as a SageMaker processing job, in this case a PySpark processor.

There are 3 modes that the compaction util supports:

- **full** : This mode will compact all files till date for the given feature group.
- **incremental** : This mode will compact all files for the previous day for the given feature group.
- **day** : This mode will compact all files for the specified date for the given feature group.

The utility takes in these parameters:

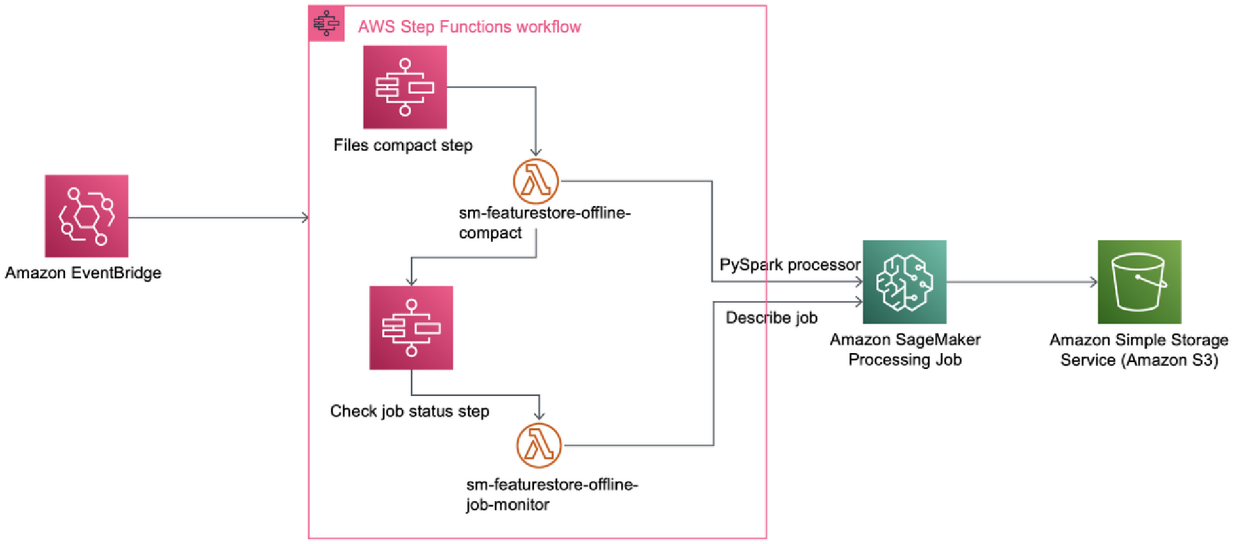
- **feature_group_name - Required**: Name of the feature group
- **compact_mode - Optional**: The mode in which compaction should be run, supported modes are 'incremental','full','day'. Defaults to 'incremental'
- **partition_mode - Optional**: Valid values are 'hour' and 'day', default is 'hour'. This specifies if the files need to be compacted for an hour, or compacted for the entire day into a single file.
- **compact_uri - Optional**: Target S3 URI where the compacted files should be stored. Defaults to a "compact-offline-store" prefix parallel to "offline" prefix in the feature store S3 bucket.
- **year, month, day - Required if mode is 'day'**: Specific date for which compaction should be run. Format should be YYYY for year, MM for month and DD for day.

Solution:

This is a serverless architecture leveraging EventBridge to fire an event on a scheduled basis, Step functions to manage and track different states, lambda functions to create and run SageMaker processing job which compacts the files.
The EventBridge is used to schedule the compaction, the first time run can be in full mode and from then can be scheduled to run in an incremental mode. The target for the EventBridge is a Step function workflow.

Step function has two main steps

- First step takes the input parameters from the incoming event, and calls sm-featurestore-offline-compact lambda function. The responsibility of the lambda function is to create and run a SageMaker processing job, giving it the python script that has logic for data compaction.
- Second step calls sm-featurestore-offline-job-monitor lambda function that polls the Processing job to see if it has completed, and if its a success or failure.



Key things to note:

- The step transitions can be monitored via the Step Function console.
- All timezones are in UTC on AWS, so the script takes UTC timezone for incremental processing. Also keep this in mind when you schedule cron job via EventBridge for incremental processing.
- If utility is run for the same files, fresh compact files will overwrite the old ones.

Sample performance metrics:

FG	Instance type	Instance count	Compact mode	Original file count	Final size	Processing Job time	Compaction time
FG 1	ml.m5.4xlarge	1	full	271	3.7 MB	5 mins	34 sec
FG 2	ml.m5.4xlarge	1	full	218 (Split across 3 days)	413 KB	5 mins	40 sec
FG 3	ml.m5.4xlarge	1	full	54	144 KB	5 mins	28 sec

Limitations:

1. Utility supports full, a particular day or incremental (day over day) modes. It does not accept a date range.
2. Utility processes minimum of a day's worth of files, it does not support incremental compaction of hourly folders. The recommendation is to run a day's worth of compaction for the prior day.

Enhancements that could be made:

1. Create a glue table and crawler for compacted feature groups.
2. Support date ranges.