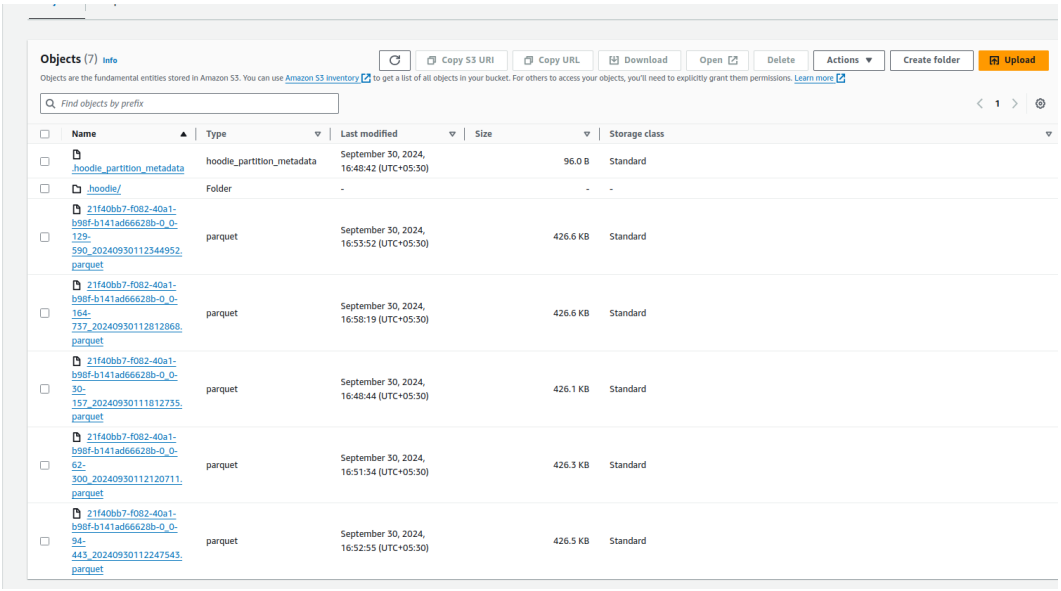
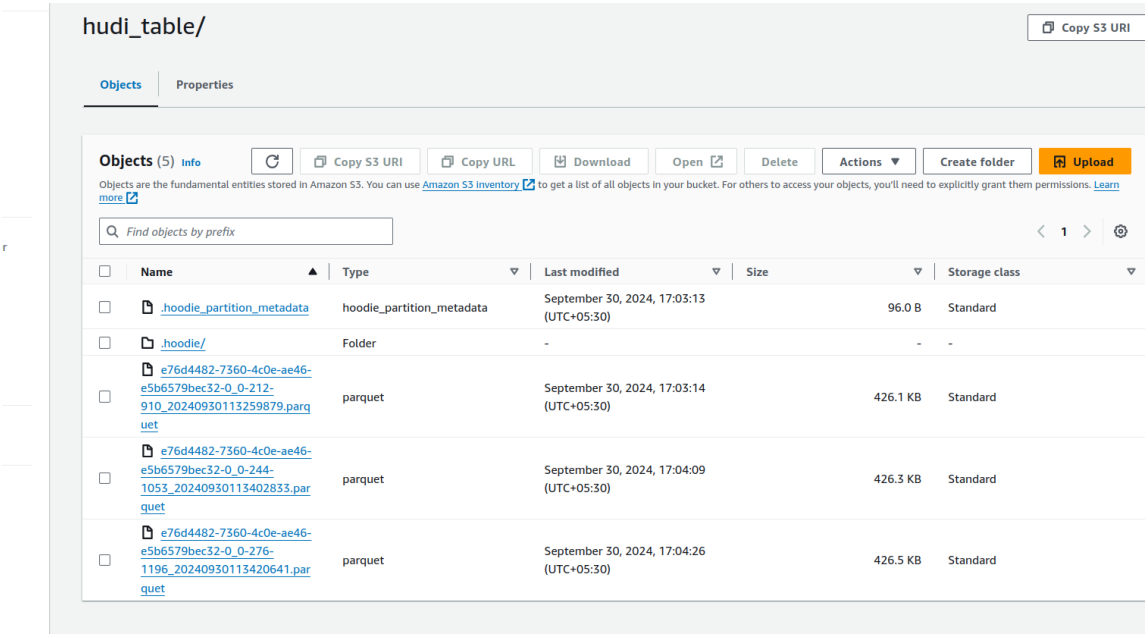
Hudi cleaner:

Before:



After adding cleaner policy:



```
# Cleaner configurations
    'hoodie.cleaner.policy': 'KEEP_LATEST_COMMITS',  # Use KEEP_LATEST_COMMITS policy
    'hoodie.cleaner.max.commits': '3',  # Keep the latest 3 commits
    'hoodie.cleaner.parallelism': '4',  # Number of parallel cleaner threads
```

Based on the Hudi options you provided, the cleaning policy is set to KEEP_LATEST_COMMITS, which means Hudi will automatically clean up older versions of data, keeping only the latest 3 commits .

Since the automatic cleaning is governed by this setting, Hudi will automatically clean the table during the commit phase (upsert or write operations) by keeping only the latest 3 commits and deleting older files.

Here's an explanation of the three cleaner configuration options:

1. 'hoodie.cleaner.policy': 'KEEP_LATEST_COMMITS'

- **Purpose**: This configuration sets the cleaning policy for the Hudi table.
- **Details**:
    - **Policy**: KEEP_LATEST_COMMITS ensures that only the latest commits are retained, and the older commits are cleaned up. This helps in managing storage by removing outdated or unnecessary data versions.
    - **Use case**: It's useful when you want to limit the number of data versions in the table, keeping only the most recent versions to optimize space.

2. 'hoodie.cleaner.max.commits': '3'

- **Purpose**: This option defines how many of the most recent commits should be retained during the cleaning process.
- **Details**:
    - **Setting**: 3 means Hudi will keep the latest 3 commits and delete data files related to older commits during the cleaning process.

- **Use case**: This ensures that even if multiple versions of the data exist, the cleaner will keep the most recent 3 versions, thus providing a rollback buffer, but not overwhelming the storage with too many versions.

3. 'hoodie.cleaner.parallelism': '4'

- **Purpose**: This setting controls the level of parallelism during the cleaning process.
- **Details**:
  - **Setting**: 4 means that Hudi will run 4 cleaner threads in parallel.
  - **Impact**: Higher parallelism increases the speed of cleaning, especially for large datasets, by cleaning multiple partitions or files at the same time. However, setting parallelism too high may cause more resource contention.
  - **Use case**: This is useful in environments with sufficient resources (e.g., a multi-core machine or distributed cluster) to speed up the cleaning process without overwhelming the system.

In Summary:

- KEEP_LATEST_COMMITS: Defines the strategy to retain only the latest commits.
- max.commits = 3: Retains only the last 3 commits, helping manage storage efficiently.
- parallelism = 4: Speeds up the cleaning process by using 4 concurrent threads.

These settings together help balance version retention and efficient cleaning in a resource-optimized manner.