Snapshot and Restore Flow - DuraCloud to Chronopolis

Snapshot

1. In DuraCloud, content is added by the user into a Snapshot Storage Provider space. This is a staging area that is backed by S3.
2. The user selects a button in the space to create snapshot and enters snapshot metadata
3. The DuraCloud UI calls the storage provider snapshot task indicating space to snapshot
4. The snapshot task creates snapshot properties file and stores it in snapshot space
    a. When the snapshot properties file is added, the space is transitioned to read-only
    b. Snapshot properties file includes the depositor ID as well as details about the snapshot (account ID, space ID, date, etc)
5. The snapshot task calls to the bridge application to indicate that a snapshot needs to be taken, providing DuraCloud host/port/space.
6. The bridge application adds an entry to the snapshot db table with the details of the snapshot action
7. The bridge application connects to DuraCloud and copies all content from DuraCloud space to bridge storage
    a. During transfer, content properties are captured in a file
    b. During transfer, each content item is added to content db table (with snapshot id)
8. The bridge application creates two manifest files (md5 and sha256) for the content and verifies all content was transferred correctly
9. The bridge application sends a notification (email) to Chronopolis that a snapshot is ready
    a. Chronopolis Intake service polls for new snapshots to process
10. Chronopolis Intake service uses the content in bridge storage to construct a bag for deposit
    a. The Intake service validates content against the manifest written by the bridge application
    b. The Intake service creates the necessary bag files (bagit, bag-info, etc) that are included in the bag
11. Chronopolis Ingest service performs replication to the appropriate Chronopolis nodes
    a. The Ingest service creates replication requests to the selected replication nodes.
    b. rsync is used between Chronopolis nodes to transfer content
12. Chronopolis makes a call to the bridge application to indicate that content has been successfully copied to preservation storage
    a. Intake service checks for existing snapshots to see if they could be completed.
    b. Intake service calls bridge to update history as each node replication completes
13. The bridge application deletes the directory in bridge storage used for the snapshot
14. The bridge application makes a call to a task in the DuraCloud Snapshot Storage Provider to indicate that it is now time to clean up the snapshot content
15. The cleanup task sets a policy on the underlying S3 bucket which causes the content to be removed within 24 hours
16. The bridge application watches the snapshot space, and when it becomes empty, calls the snapshot complete task, which clears the S3 bucket policy
17. The bridge application notifies the user who requested the snapshot that it has been completed

Snapshot Display

1. A DuraCloud user selects the Snapshot Storage Provider in the DuraCloud UI
2. The DuraCloud UI (DurAdmin) makes the usual call to get spaces and also calls a task in the Snapshot Storage Provider to request a list of snapshots
3. The snapshots task calls the bridge application to request a list of snapshots
    a. Note: DuraCloud is not aware of the snapshot database. All communication with the db goes through the bridge application.
4. The bridge application queries the database (snapshot table) to retrieve a listing of snapshots which are visible to the given DuraCloud account
5. The bridge application returns list of snapshots, which are passed back up the chain to the UI
6. The DuraCloud UI displays the list of snapshots alongside the traditional spaces in a way the distinguishes the two sets, providing a Restore button on each snapshot space
7. When a snapshot space is selected, calls to another set of tasks in the Snapshot Storage Provider are made to retrieve the listing of content items, the snapshot details, and snapshot history
8. The snapshot tasks call the bridge application to request the snapshot details, history, and content list
9. The bridge application queries the database to retrieve the snapshot information
10. The bridge application returns the snapshot information, for the content list and snapshot history lists, only the first X items are returned, which are passed back up the chain to the UI
11. The DuraCloud UI displays the snapshot information in the same way it would display content items and details for a traditional space
12. Requests for more items in the content listing or snapshot history follow the same pattern, but with a parameter to indicate offset

Restore

1. The DuraCloud user browses the snapshot listings and clicks the "Request Restore" button on a snapshot
2. The DuraCloud UI calls a DuraCloud Request Restore task, which makes a call to the bridge. The bridge sends email notifications to DuraSpace staff to notify that a restore has been requested the the relevant details.
3. DuraSpace staff verifies with depositor that a restore is needed, then uses the "Restore" button on a snapshot to initiate restore.
4. The DuraCloud UI calls a DuraCloud Restore task
5. The DuraCloud Restore task creates a space where the restored content will be placed and calls the bridge application with a restore request
    a. A bucket policy will be added to the space to delete content after a set time period
6. The bridge application adds an entry to the restore db table
7. The bridge application creates a directory in bridge storage and sends a notification to Chronopolis to request a restore action
8. Chronopolis copies the contents of the snapshot bag(s) to the bridge storage directory
    a. Chronopolis will validate the files on the bridge storage against the manifest in the bag (originally created during the snapshot phase)
    b. BagIt files are omitted during the copy, leaving only the files which were originally written by the bridge application
9. Chronopolis makes a call to the bridge to indicate that content has been restored to the expected bridge storage location.
10. The bridge application verifies restored content against snapshot manifest file and against database listing (ensuring content is consistent with original snapshot data set)
11. The bridge application copies content from bridge storage to DuraCloud space
12. The bridge application reads the content metadata file and updates each content item with its metadata values

13. The bridge application verifies that content transferred to DuraCloud is consistent with the content in bridge storage
14. The bridge application deletes the directory in bridge storage used for the restore action
15. The bridge application notifies the user who requested the restoration that the process is complete, informs them of the space ID where they can find their content, and tells them the date on which the content will expire (and be deleted.)

Database Tables

DB

- Table to capture spring batch processing (states)
- Table to capture snapshots
    - Snapshot ID
    - Snapshot date
    - Source DuraCloud Host
    - Source DuraCloud Port
    - Source DuraCloud Space ID
    - Source DuraCloud Store ID
    - Number of content items in snapshot
    - List of duracloud accounts for which the snapshot is visible
    - Snapshot status
- Table to capture content items
    - Snapshot ID
    - Content ID
    - Content Metadata List (serialized)
- Table to capture restore actions
    - Snapshot ID
    - Restoration ID
    - DuraCloud Host
    - DuraCloud Port
    - DuraCloud Space ID
    - DuraCloud Store ID
    - Start Date
    - End Date
    - Restore status

Interface Points - APIs

Note: Colors indicate where calls to a particular interface method will originate. For example, a method in the bridge API is green if the calls to that method are made from a Snapshot Storage Provider Task. A method in the bridge API is red if the calls to that method are made from Chronopolis.

DuraCloud UI (DurAdmin)

- <No additional API calls>

Snapshot Storage Provider Tasks

- Create snapshot (spaceID, storeID)
  - create snapshot properties file, store it in space
  - space transitioned to read only
  - call made to bridge API to create snapshot
- Get snapshot status (snapshot ID)
  - call made to bridge for status
- Snapshot cleanup (spaceID)
  - set bucket deletion policy
- Snapshot complete (spaceID)
  - remove bucket deletion policy
- Get list of snapshots
  - call to bridge for snapshot list, formats results
- Get list of content items (snapshot ID, offset, maxresults)
  - call to bridge for content item list
- Restore snapshot (snapshot ID, storeID)
  - verify that a restore is not already in place
  - create a space for the snapshot to be placed into
  - call to bridge to restore
- Get restore status (snapshot ID)
  - call made to bridge for status

Bridge API

- Create snapshot (host, port, spaceID, storeID, snapshotID)
  - create snapshot db entry
  - copy content to bridge storage
  - create a manfitest for content
  - notify chronopolis that snapshot is ready for storage
- Get snapshot details (host, port, snapshot ID)
  - query snapshot status from the db (and from chronopolis, depending on status)
- Snapshot complete (snapshot ID, alternate IDs)
  - delete content in bridge storage
  - call task to indicate that snapshot is complete
  - notify end user that snapshot is complete
- Get list of snapshots (host)
  - Queries db for snapshot list available to host
- Get list of content items (snapshot ID, offset, maxresults)
  - Queries db for content item list for the given snapshot ID
- Add Snapshot History (snapshotID or alternateID, history details)
  - Attaches a new history event to a snapshot
- List Snapshot History (snapshotID, pageNumber, pageSize)
  - Queries db for all history events associated with a snapshot
- Restore snapshot (host, port, spaceID, storeID, snapshot ID)
  - create restore db entry
  - create a directory on bridge storage
  - requests restore from chronopolis
- Restore complete (restoration ID)
  - copy content to Duracloud
  - update content metadata
  - verify restored content
  - delete content on bridge
  - notify end user that restore is complete
- Get restore details (restoration ID)
  - query restore status from the db (and from chronopolis, depending on status)

Chronopolis API

- Store snapshot (snapshot ID, bridge storage path)
- Get snapshot status (snapshot ID)
- Restore snapshot (snapshot ID, bridge storage path)
- Get restore status (snapshot ID)