SCOPE OF WORK

SOW NO: 5  EFFECTIVE DATE: 8/20/2021

**1. Detailed description of project to be accomplished by Cloud Operations Engineer:**

**Cloud Operations will provide Navaneethan Gunasekaran for this SOW in support of OPADM Ascension Delivery project**

We need a solution to deliver files to clients from AWS. I have divided this into 3 portions. The most time sensitive part is the Ascension s3 to Google Cloud Storage. This needs to handle moving large files monthly from our environment to their Google Cloud. The second part and third parts are options to handle delivering files to the rest of our OPADM consumers.

Note: These 3 parts can be combined (as much as possible) into one solution that is configurable by the DEXTR team to add and remove clients and choose the method of delivery within that configuration.

**Requirements**

- **Part 1: Delivery to Google Cloud**
- **Part 2: Delivery to external SFTP**
- **Part 3: Drop file in internal S3 (path tbd)**

Existing process scripts for reference: opadm github

Configuration file:

- Configuration Name
    - Client h group
    - Contact email address (multiple addresses excepted)
    - Send notifications (Boolean)
    - Pick up location
        - *What S3 info is needed here?*
    - Destination
        - Google cloud
            - *What GC info is needed here?*
            - *Reference to secret manager?*
        - SFTP
            - *What SFTP info is needed here?*
            - *Reference to secret manager?*
        - S3
            - *What S3 info is needed here?*
    - Bypass file checks (allows for sending same file again if needed)
    - Trigger on [H NUMBER].done file (Boolean)

- *Anything else needed to make this function that I didn't think of, suggested by CCOE*

Format of Dynamo DB:

- Partition Key: Client H number
- Sort Key: etag of file (or another unique file identification i.e. md5 checksum)
- Attributes:
  - timestamp (updated at time of creation or attr. change)
  - status (string, processing,success,fail)
  - filePath (string, path to file)
  - sendAttempts (int)
  - active (boolean)
  - fileName (string, name of file)
  - *Anything else needed to make this function that I didn't think of, suggested by CCOE*

START PROCESS

Time based Trigger, every 15 minutes

- Health check
  - Check Dynamo DB for any entry where active = true
    - Update log with those entries as an error message
    - Change active to false
- For each configuration:
  - check configured s3 pick up path
  - If *trigger on done file* = true
    - Proceed when [H NUMBER].done file is found
    - Delete [H NUMBER].done file
    - Read all available files in the s3 pick up and… (to next step)
  - If *trigger on done file* = false
    - Read all available files in the s3 pick up and… (to next step)

For each file

- If *bypass file checks* = false
  - Check against dynamo db etags to see if we already sent that file
  - If file matches an entry in the db
    - Delete the file from the S3 pick up
  - If file is not found in the db
    - Add the file to the db with the attributes:
      - status = processing
      - sendAttempts = 0
      - timestamp = current time
      - active = true
      - filePath = full path of file

                - fileName = name of file
- If *bypass file checks* = true
    - Check against dynamo db etags to see if we already sent that file
    - If file matches an entry in the DB
        - Update the attributes:
            - status = processing
            - sendAttempts = 0
            - timestamp = current time
            - active = true
            - filePath = full path of file
            - fileName = name of file
    - If file is not found in the db
        - Add the file to the db with the attributes:
            - status = processing
            - sendAttempts = 0
            - timestamp = current time
            - active = true
            - filePath = full path of file
            - fileName = name of file

Step function calls correct Fargate container based on client H number destination

Update log with job start info (time, H number, files, status)

If Send notifications = true

- Use dynamo db to populate the 'Job started' email
- For all entries matching the H number and active = true include:
    - Client H#
    - Time job started (current time)
    - File Names
    - File Status
- Send the email to the addresses in the configuration file

Check Dynamo DB for all entries matching the H number and active = true

- Pass all file paths to aws batch job

All files are delivered to specified destination in the configuration file

For each file

- on successful transfer
    - update the dynamo db
        - status = success
        - timestamp

- on failure
    - update the dynamo db
        - increment send attempts
        - timestamp

Check dynamo db for all entries matching the H number and that active = true

- For all entries that status = processing:
    - retry sending. Send back to aws batch job.
    - if send attempts = 3
        - update status = fail

Update log with job finish info (time, H number, files, status)

If Send notifications = true

- Use dynamo db to populate the 'Job finished' email
    - For all entries matching the H number and active = true include:
        - Client H#
        - Time job finished (current time)
        - File Names
        - File Status
- Send the email to the addresses in the configuration file

Update dynamo db for all entries matching the H number and active = true

- change active to false

END PROCESS

Notes:

- If the module is destroyed/spun down, the dynamo DB data will need to persist.
- Only one instance of this process should exist at a time
- If the process restarts due to finding a 'processing' status, the client/stake holders should not be notified. One email goes out at the very beginning of the process, and one goes out after the max retry has been met (or on all successful after one run)

**2.** Deliverables to be produced by Cloud Operations Engineer

*List all deliverables to be produced by Cloud Operations Engineer and describe them with as much detail as possible*

- **Part 1:** AWS architecture build in Platform DFE account
- **Part 2:** AWS architecture build in Platform DFE account
- **Part 3:** AWS architecture build in Platform DFE account

**3.** Estimated date of completion for each deliverable

1. 9/30/21
2. 10/15/21
3. 10/15/21

**4.** Acceptance criteria for each deliverable

Solution is cost effective, reliable, and fast. All above requirements met.

**5.** Time spent on meetings

*Describe the different meetings the Cloud Operations engineer will need to attend and the duration*

1. We have a recurring check in on Tuesday morning and Thursday morning that can be used for questions and updates.

SOW Created By: Kushal Naidu

SOW Created Date: 8/20/21

SOW Accepted By:

SOW Accepted Date: