SOFTWARE REQUIREMENTS SPECIFICATION

for

AWS Glacier wrapper

Version 1.0 unapproved

Prepared by <Ankur Singh>

Personal Project

November 3, 2019

Contents

Revision History

Name	Date	Reason For Changes	Version

1 Introduction

1.1 Purpose

This tool provides an interface to archive local files to AWS Glacier and provides tools to upload, download and query the files present in archive.

1.2 Intended Audience and Reading Suggestions

User targeted for this tool are those who are familier with command line tools and basics of AWS services. User is expected to setup environment in order to access AWS Glacier service.

1.3 Project Scope

1.3.1 Goal

To provide easy interface for an user to archive local files to AWS Glacier.

1.3.2 Benefit

It will insulate user from the the nuances of handling file upload and download from archives. Right now the user can not query the metadata of files uploaded in Glacier.This tool provide easy interface to query such data along with being able to upload or download files.

2 Overall Description

2.1 Product Perspective

As far as known by research, such tool is not present. There are some tools which can help in archiving the data, but query tools for archived data are not present. This tool want to simplify archiving process by providing an interface which feels like local storage.

2.2 Product Functions

- **Upload** Upload a list of files or folders.
- **Download** Download the list of files or folders.
- **Delete** Delete the list of files or folders.
- **Sync** Tool will maintain a local database of the status of the files. This function will sync local machine state with the latest AWS glacier state.
- **Query** Query the status of the files in the archive.

2.3 User Classes and Characteristics

User should have some programming experience. User is expected to setup AWS Glacier service access environment in the local machine. Also user should be able to execute a python script and use command line tool.

2.4 Operating Environment

This tool has unit system as target. It will be developed and tested on Ubuntu 16.04 version. It is expected to run on all major unix like systems. But it will not be tested for such gurantee of performance on other operating systems.

2.5 Design and Implementation Constraints

- There is not gurantee for cost minimization. Since it is an archiving tool, it is expected that the user knows that the download and modification operation can be costly if run frequently. However query operation will not use AWS service

directly (Apart from optional sync operation), hence it does not incur any extra charge.

- It does not handle AWS Glacier management.It is expected that Glacier management is handled separatly.

2.6 Assumptions and Dependencies

- **Operating System** Ubuntu 16.04 or above.
- **Dependencies** boto3, python3, pip3, aws-cli
- **Other** Programmatic access permission to AWS Glacier service

3 External Interface Requirements

3.1 User Interfaces

This is a command line tool. It is not installable distribution,i.e. the user is required to run a single executable script file and interact with the tool using command line arguments.

4 System Features

4.1 Archive and upload a list of files

Upload a list of files or directories.

4.1.1 Description and Priority

High priority User will provide a list of files or directories names. Name can be absolute or relative path. Tool will archive in particular format (file_name.tar.gz) and upload to the vault in Glacier. If operations fails for some files then provide the error message in the log.

4.1.2 Stimulus/Response Sequences

User: provides list of file names.
Tool: If some file is not recognised in the system, provide warning but for the rest of the recognised files, archive and upload them. In case of any failure (unable to recognize the file name, unable to archive some file, unable to upload etc.), the tool will provide error messages in the log and continue with the operation.

4.1.3 Functional Requirements

REQ-1: Take user input
Interface for user input will be ./script [options] valut-name file-list...
REQ-2: Handle the case that no file is provided or no file name is recognised
If no file is provided, output one error message and abort the operation.
REQ-3: Some file names are not recognised
Provide error message in error logs with the file names and continu with the operation.
REQ-4: Unable to archive some files
Provide error message in error logs with the file names and continu with the operation.
REQ-5: Archive operation failed
Provide error message and abort the operation.
REQ-6: Upload operation failed
Provide error message and abort the operation.
REQ-7:Upload is taking too much time.
If upload takes more than some specific time then abort uploading rest of the files and provide the list of the files not uploaded. **REQ-7**:Problem with AWS Glacier service
Provide the relevant error information and abort the operation.

4.2 Download the list of files

5 Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

6 Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>