1. How to check your podcast episode download statistics

[5/20/2020 6:22 PM CST]

1.1. [8/9/2020 2:46 PM CST]

Check your email, everyday!

The podcast-stats reporter software program running on AWS Lambda emails a podcast episode download statistics report once everyday using AWS SES.

2. How to configure the podcast AWS S3 bucket for download statistics reporting

2.1. [5/24/2020 11:03 AM CST]

[5/25/2020 1:15 AM CST]

1. Turn on server access logging on the podcast episodes AWS S3 bucket(s).
2. Set up AWS Lambda for podcast stats auto-emailing.

(those are all!)

3. how to set up AWS Lambda and install the podcast-stats emailing functions

[6/18/2020 8:21 PM CST] process spec

1. create an IAM policy for the podcast-stats emailing AWS Lambda functions, with relevant AWS S3 objects management and AWS SES email sending.
    1.1. In the IAM console, choose Policies on the left pane.
    1.2. Choose Create policy.
    1.3. Choose the JSON tab.
    1.4. Copy and paste the IAM policy JSON template.
    1.5. Modify the JSON policy according to your needs.
    1.6. Choose Review policy.
    1.7. Name the IAM policy, PodcastStatsReporting.
    1.8. Choose Create policy.
2. create an IAM role for the AWS Lambda function.
    2.1. Open the roles page in the IAM console.
    2.2. Choose Create role.
    2.3. Create a role with the following properties.
        2.3.1. Trusted entity – Lambda.
        2.3.2. Permissions – AWSLambdaBasicExecutionRole, PodcastStatsReporting.

        2.3.3. Role name – podcast-stats-reporting-lambda-role.
3. create the AWS Lambda function.
    3.1. Open the Lambda console.
    3.2. Choose Create function.
    3.3. Configure the following settings:
        3.3.1. Name – podcast-stats-reporter.
        3.3.2. Runtime – Python 3.8.
        3.3.3. Role – Choose an existing role.
        3.3.4. Existing role – podcast-stats-reporting-lambda-role.
    3.4. Choose Create function.
    3.5. To configure a test event, choose Test.
    3.6. For Event name, enter test.
    3.7. Choose Create.
    3.8. Set the Lambda function timeout to 5 minutes (300 seconds). (it may take longer to execute than the default 3 seconds.)
    3.9. Add the required Lambda function environment variables.
    3.10. To execute the function, choose Test.
4. test run the AWS Lambda function.
5. check the AWS Lambda function log to make sure that the function has worked without an error.
6. configure the daily auto-execution of the podcast-stats reporting software on AWS Lambda.
    6.1. (specify here exactly how!)
    6.2. https://docs.aws.amazon.com/lambda/latest/dg/services-cloudwatchevents.html
    6.3. "`cron(15 4 * * ? *)`" [4:15 AM (UTC) every day]
7. set automatic deletion on the AWS-Lambda AWS CloudWatch logs.
    7.1. (specify here exactly how!)

(if needed or wanted, delete any unnecessary AWS-Lambda AWS CloudWatch metrics.)

[6/9/2020 11:41 AM CST]
follow the instruction at **https://docs.aws.amazon.com/lambda/latest/dg/lambda-python.html**.

https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

for clean up, follow the instruction at **https://docs.aws.amazon.com/lambda/latest/dg/getting-started-create-function.html**.

3.1. IAM policy JSON

3.1.1. [9/4/2020 9:13 AM CST]

```
{
   "Version": "2012-10-17",
```

```
  "Statement": [
     {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": [
           "arn:aws:s3:::podcast-episodes",
           "arn:aws:s3:::podcast-episodes/*",
           "arn:aws:s3:::data-private",
           "arn:aws:s3:::data-private/podcast-logs",
           "arn:aws:s3:::data-private/podcast-logs/*",
           "arn:aws:s3:::data-private/podcast-stats",
           "arn:aws:s3:::data-private/podcast-stats/*"
        ]
     },
     {
      "Effect":"Allow",
      "Action":[
       "ses:SendEmail",
       "ses:SendRawEmail"
      ],
     "Resource":"*"
     }
  ]
}
```

4. software process outline

[6/9/2020 7:33 AM CST]

Podcast-stats emailing Python code outline
-get the list of all the files in the log AWS S3 bucket.
-process the bucket objects with the creation or modification time between the report period range. This will work correctly, since each log file contains log entries up to its creation or modification time, never beyond.
-read the content of each applicable log file, use a precompiled regexp to parse each line, and use only the log entries with applicable operation type (GET) and time (i.e. within the reporting range). Using a Python dictionary, keep the count of each key or file name in the dictionary. Test and see if bucket names include folder names too. If they do, dictionary (bucket name to dictionary) to dictionary (file name to count) data structure will have to be used.
-using the count dictionary contents and AWS SES API, generate and send the podcast stats report email. Use HTML email for professional, responsive email contents.
(That's all, folks!)
(UPDATES processing to the latest file, since a later file can contain earlier log entries. Also, process several hours after the end date and time, since it can take up to a few hours for a log file to be generated. If the deadline is 12am for including the log entries earlier than that time and

date, the processing should be done at 3 or 4 am. Create and use the Python time objects for time comparison.)

5. local-machine code test procedure

[7/2/2020 10:24 AM CST]

5.1. test command

[7/28/2020 11:38 AM CST]

```
python.exe aws_podcast_stats_reporter.py "PODCAST_NAMES_IN_STATS_REPORTS" \
"PODCAST_EPISODE_AWS_S3_BUCKET_FOLDER_PATHS" \
"PODCAST_LOG_AWS_S3_BUCKET_FOLDER_PATHS" \
"PODCAST_STATS_AWS_S3_BUCKET_FOLDER_PATHS" \
"FIRST_COLUMN_DATA_TYPE_IN_EMAIL" \
"PODCAST_STATS_SENDER_EMAIL_ADDRESS" \
"PODCAST_STATS_RECIPIENT_EMAIL_ADDRESS" \
"DAILY_PODCAST_STATS_EMAIL" "WEEKLY_PODCAST_STATS_EMAIL" \
"MONTHLY_PODCAST_STATS_EMAIL" \
"INCLUDE_NUMBERS_OF_RETRIES_IN_PODCAST_STATS_EMAILS" \
"SEPARATE_PODCAST_STATS_EMAILS" \
"PODCAST_STATS_HTML_EMAIL_BACKDROP_COLOR" \
"PODCAST_STATS_HTML_EMAIL_SECTION_HEADING_BACKDROP_COLOR" \
"PODCAST_STATS_HTML_EMAIL_SECTION_HEADING_FONT_NAME" \
"PODCAST_STATS_HTML_EMAIL_SECTION_HEADING_FONT_SIZE" \
"PODCAST_STATS_HTML_EMAIL_SECTION_HEADING_FONT_COLOR" \
"PODCAST_STATS_HTML_EMAIL_BODY_FONT_NAME" \
"PODCAST_STATS_HTML_EMAIL_BODY_FONT_SIZE" \
"PODCAST_STATS_HTML_EMAIL_BODY_FONT_COLOR" \
"NON_DOWNLOAD_ENTRY_PODCAST_LOG_AWS_S3_BUCKET_OBJECTS_DELETIO
N" \
"IRRELEVANT_DOWNLOAD_ENTRY_PODCAST_LOG_AWS_S3_BUCKET_OBJECTS_
DELETION" \
"RELEVANT_DOWNLOAD_ENTRY_PODCAST_LOG_AWS_S3_BUCKET_OBJECTS_DE
LETION" "PODCAST_STATS_REPORT_LANGUAGE"
```