

## Part 1 – The Public Cloud is ideal for data processing

Public cloud is the next technology that will be adopted by everyone in time. The technology era is slowly ending, while the cloud era is slowly rising prominence, not only due to the cost of building hardware infrastructure but also the perceivable delay that is apparent when using physical data warehousing. The covid-19 pandemic has made it more apparent about the requirements of easy access to data, that is more secure than transferring files and the need for collaboration over the internet.

Many enterprises are slowly converting towards a cloud infrastructure to not only reduce the current cost of running the business but also to reach out to a farther market and customers that are stuck in their homes. Public cloud has also led to a boom in remote work jobs as it keeps the leisure of life safe, while keeping the efficiency same, or sometimes even better.

The public cloud being the next thing leads to the question of data processing which has also been in the rise in the recent years, the requirement of large datasets that must be stored and processed for hours for a chance of a proper prediction or pattern outcome among data scientists has been apparent. The resources required to run proper algorithms is nothing to scoff at, the importance of CUDA in complex machine learning algorithms has become essential, which for research or learning is a costly endeavor. The public cloud and the services it can provide through various cloud platforms ease up the issue of incurring costs in advance, as marketed by various cloud platforms, the services provided are pay per use, and will become cheaper the more people use it as the cost for the data warehouse has already been incurred and repaid. It also saves up the idea of constant upgrading of different peripherals to match the current technology.

The AWS services that make it worthwhile and apparent that public cloud is ideal for data processing include AWS compute optimize, AWS autoscaling, AWS S3, AWS X-Ray, AWS Glue. While AWS X-Ray and AWS Glue are the new services that have been provided for application management, monitoring and optimization, while Glue is a dream come true for doing data analytics on the cloud. The long existing services like compute optimize, auto scaling and S3 make cloud a stable choice, the auto scaling and compute optimize makes sure you do not run out of resources while your code is running and algorithm, which has happened to me more times than I'd like to admit. While S3 makes sure you don't run out of storage while using API to keep getting the latest data for proper analysis of a trend, a stock, or any real-time data that exists globally.

There are a few instances where the public cloud does feel effy, including the issues with data-usage and cookie-usage by different companies, the issue with data privacy, which lead to some situations where data processing can do more harm than good. One of the primary reason is lack of competition if you get trapped under one cloud platform with no way to shift to another without incurring massive changes in personnel and relearning the entire framework, data processing of data that can easily lead to privacy leaks and social networking data processing which has caused enough problems world-wide, in politics and fake news.

The public cloud is ideal for data processing if utilized with proper care and set guidelines for the user and the provider.

## Part 2 Scaling the WordFreq Application

### Tasks-

- > Install the application
- > Design and Implement Autoscaling
- > Perform Load Testing
- > Optimize the Wordfreq architecture
- > Final Repot

### Report -

The application calculates the frequency of words in a text file. It considers all the unique words in the text file separated by `_space_` and if any word is repeated that is added to the count, the final calculation is done at the end of the file and then it is sorted before given the printout. The file is read the queue service one at a time and stored into DynamoDB. The DynamoDB table does not sort the values, it stores all values as they are sorted when the output is to be displayed to the user.

The application considers only plain text files and does not take complex characters in the filename .

The autoscaling configuration I used to implement were t1.micro and t3 micro instances, over the existing free tier t2.micro that can be changed depending on the day and work, to speed through the process of changing between these instances, they were stored as launch instance types and could convert the auto scaling groups as per the requirements.

## Dynamic scaling policies (1) [Info](#)

### Target Tracking Policy



Policy type:

Target tracking scaling

Enabled or disabled?

Enabled

Execute policy when:

As required to maintain Average CPU utilization at 50

Take the action:

Add or remove capacity units as required

Instances need:

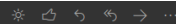
300 seconds to warm up before including in metric

Scale In:

Enabled

Successful	Launching a new EC2 Instance: i-0cdd7040b072e59ea	At 2022-01-12T08:06:11Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2022 January 12, 08:06:13 AM +00:00	2022 January 12, 08:06:31 AM +00:00
Successful	Terminating EC2 Instance: i-0d72cdef0bde9058e	At 2022-01-12T08:06:01Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2022 January 12, 08:06:01 AM +00:00	2022 January 12, 08:11:21 AM +00:00
Successful	Launching a new EC2 Instance: i-0d72cdef0bde9058e	At 2022-01-12T08:04:10Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2022 January 12, 08:04:13 AM +00:00	2022 January 12, 08:04:29 AM +00:00
Successful	Terminating EC2 Instance: i-00e10859514b28016	At 2022-01-12T08:04:00Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2022 January 12, 08:04:01 AM +00:00	2022 January 12, 08:09:18 AM +00:00
Successful	Launching a new EC2 Instance: i-00e10859514b28016	At 2022-01-12T07:52:10Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2022 January 12, 07:52:13 AM +00:00	2022 January 12, 07:52:30 AM +00:00
Successful	Terminating EC2 Instance: i-00a0607225d9675f1	At 2022-01-12T07:52:00Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2022 January 12, 07:52:01 AM +00:00	2022 January 12, 07:57:27 AM +00:00
Successful	Terminating EC2 Instance: i-0f28b6994c63656c7	At 2022-01-12T07:22:42Z a monitor alarm TargetTracking-wordfreq-autos-group-AlarmLow-717c9f86-d1c0-47c3-b1e6-f8580c0e0b85 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 2 to 1. At 2022-01-12T07:22:50Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2022-01-12T07:22:50Z instance i-0f28b6994c63656c7 was selected for termination.	2022 January 12, 07:22:50 AM +00:00	2022 January 12, 07:28:43 AM +00:00
Successful	Launching a new EC2 Instance: i-00a0607225d9675f1	At 2022-01-12T07:07:04Z a monitor alarm TargetTracking-wordfreq-autos-group-AlarmHigh-3fffb58c-8bbf-4251-845a-e29cf65ae821 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 1 to 2. At 2022-01-12T07:07:09Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2022 January 12, 07:07:15 AM +00:00	2022 January 12, 07:12:44 AM +00:00
Successful	Launching a new EC2 Instance: i-0f28b6994c63656c7	At 2022-01-12T07:02:05Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2022 January 12, 07:02:08 AM +00:00	2022 January 12, 07:02:41 AM +00:00

AWS Notifications <no-reply@sns.amazonaws.com>



Wed 1/12/2022 8:06 AM

To: Akshat Pande

Service: AWS Auto Scaling

Time: 2022-01-12T08:06:30.790Z

RequestId: a455f935-4ae1-f56e-4fad-a4b1a29e09b5

Event: autoscaling:EC2\_INSTANCE\_LAUNCH

Accountid: 035036656553

AutoScalingGroupName: wordfreq-autos-group

AutoScalingGroupARN: arn:aws:autoscaling:us-east-1:035036656553:autoScalingGroup:990af58b-a68d-40a3-a62e-7867990fda43:autoScalingGroupName/wordfreq-autos-group

ActivityId: a455f935-4ae1-f56e-4fad-a4b1a29e09b5

Description: Launching a new EC2 instance: i-0cdd7040b072e59ea

Cause: At 2022-01-12T08:06:11Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.

StartTime: 2022-01-12T08:06:13.885Z

EndTime: 2022-01-12T08:06:30.790Z

Status: InProgress

StatusMessage:

Progress: 50

EC2Instanceid: i-0cdd7040b072e59ea

Details: ("Subnet ID": "subnet-03aeed090ca288fc", "Availability Zone": "us-east-1a")

Origin: EC2

Destination: AutoScalingGroup

```

Service: AWS Auto Scaling
Time: 2022-01-12T08:09:18.333Z
RequestId: 8ba5f935-42c6-8b91-2251-9591d064fe67
Event: autoscaling:EC2_INSTANCE_TERMINATE
Accountid: 035036656553
AutoScalingGroupName: wordfreq-autos-group
AutoScalingGroupARN: arn:aws:autoscaling:us-east-1:035036656553:autoScalingGroup:990af58b-a68d-40a3-a62e-7867990fda43:autoScalingGroupName/wordfreq-autos-group
ActivityId: 8ba5f935-42c6-8b91-2251-9591d064fe67
Description: Terminating EC2 instance: i-00e10859514b28016
Cause: At 2022-01-12T08:04:00Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.
StartTime: 2022-01-12T08:04:01.058Z
EndTime: 2022-01-12T08:09:18.333Z
StatusCode:InProgress
StatusMessage:
Progress: 50
EC2Instanceid: i-00e10859514b28016
Details: {"Subnet ID":"subnet-03aeed090ca288fc","Availability Zone":"us-east-1a"}
Origin: AutoScalingGroup
Destination: EC2

```

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:  
<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:035036656553:wordfreq-auto-scaled:8299f197-fec2-4a1a-8454-c21b56d1f4dd&Endpoint=tx21857@bristol.ac.uk>

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check
<input type="checkbox"/>	-	<a href="#">i-0bb2b4e5be36be39f</a>	⊖ Terminated 🔍 🔍	t1.micro	-
<input type="checkbox"/>	wordfreq-dev	<a href="#">i-0ffa983b335e10ba2</a>	⊖ Stopped 🔍 🔍	t2.micro	-
<input type="checkbox"/>	-	<a href="#">i-0e53dbf0315e604b5</a>	⊖ Stopped 🔍 🔍	t1.micro	-
<input type="checkbox"/>	-	<a href="#">i-0fda1592a11bb74bf</a>	✔ Running 🔍 🔍	t1.micro	🕒 Initializing

#### Instances (3) [Info](#)

🔍 Search

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check
<input type="checkbox"/>	-	<a href="#">i-0bb2b4e5be36be39f</a>	✔ Running 🔍 🔍	t1.micro	✔ 2/2 checks passed
<input type="checkbox"/>	wordfreq-dev	<a href="#">i-0ffa983b335e10ba2</a>	✔ Running 🔍 🔍	t2.micro	✔ 2/2 checks passed
<input type="checkbox"/>	-	<a href="#">i-0e53dbf0315e604b5</a>	✔ Running 🔍 🔍	t1.micro	✔ 2/2 checks passed

```

Jan 12 05:36:07 ip-172-31-27-24 wordfreqservice[899]: Processing message b55f7981-9ef1-4488-bb3e-1a2641eea8f5
Jan 12 05:36:07 ip-172-31-27-24 wordfreqservice[899]: Worker 0 received job b55f7981-9ef1-4488-bb3e-1a2641eea8f5
Jan 12 05:36:17 ip-172-31-27-24 wordfreqservice[899]: Received job result b55f7981-9ef1-4488-bb3e-1a2641eea8f5
Jan 12 05:36:17 ip-172-31-27-24 wordfreqservice[899]: Successfully processed job b55f7981-9ef1-4488-bb3e-1a2641eea8f5
Jan 12 05:36:17 ip-172-31-27-24 wordfreqservice[899]: Deleted message, b55f7981-9ef1-4488-bb3e-1a2641eea8f5

```

It was also dynamically scaled based on the health of the instances and CPU utilization, after doing multiple test loads and checking networks, the CPU utilization was around 40% for file uploads of around 150mb with all files having separate unique data. Hence, the dynamic settings were brought down to 30% to avoid halting due to instance timeout times (errors occurred in run\_worker.sh journal) and added load balancer in case of multiple parallel uploads to the Wordfreq Application.

Target Tracking Policy

☐

Policy type:  
Target tracking scaling

Enabled or disabled?  
Enabled

Execute policy when:  
As required to maintain Average CPU utilization at 25

Take the action:  
Add or remove capacity units as required

Instances need:  
300 seconds to warm up before including in metric

Scale In:  
Enabled

Launch configuration

Edit

Launch configuration wordfreq-launchconfig	AMI ID ami-05e4673d4a28889fe	Security groups sg-0ab5470a0035a927f <a href="#">🔗</a>
Instance type t3.micro	Key pair name learnerlab-keypair	Create time Wed Jan 12 2022 08:22:23 GMT+0000 (Greenwich Mean Time)
Storage (volumes) /dev/sdc		

[View details in the launch configuration console](#) [🔗](#)

## Building and Testing

After following the guidelines to setup, the app, the first few steps were to setup the auto-scaling configuration and the auto scaling groups that would be applicable to the wordfreq application.

I primarily tested the wordfreq application with the default settings and the backed-up AMI image with the same instance for testing out the application. The delay of 10 seconds aside, there were inconsistencies during the testing when going through the journal and the SQS, messages in the wordfreq-job were being restored back due to instance timeouts and exceptions in filenames took the entire process into a slow procedure throughout all files with failed messages.

Brave Web Browser Jan 12 07:46

console.aws.amazon.com/sqs/v2/home?region=us-east-1#/queues

Queue wordfreq-results has been purged successfully.

Amazon SQS > Queues

Queues (2)

Search queues by prefix

Name	Type	Created	Messages available	Messages in flight
wordfreq-jobs	Standard	11/01/2022, 11:22:53 GMT	0	0
wordfreq-results	Standard	11/01/2022, 11:24:23 GMT	70	0

Brave Web Browser Jan 12 07:45

console.aws.amazon.com/sqs/v2/home?region=us-east-1#/queues

Queue wordfreq-results has been purged successfully.

Amazon SQS > Queues

Queues (2)

Search queues by prefix

Name	Type	Created	Messages available	Messages in flight
wordfreq-jobs	Standard	11/01/2022, 11:22:53 GMT	0	5
wordfreq-results	Standard	11/01/2022, 11:24:23 GMT	62	0

Brave Web Browser Jan 12 07:35

console.aws.amazon.com/sqs/v2/home?region=us-east-1#/queues

Queue wordfreq-results has been purged successfully.

Amazon SQS > Queues

Queues (2)

Search queues by prefix

Name	Type	Created	Messages available	Messages in flight
wordfreq-jobs	Standard	11/01/2022, 11:22:53 GMT	33	6
wordfreq-results	Standard	11/01/2022, 11:24:23 GMT	8	0

Brave Web Browser Jan 12 07:33

console.aws.amazon.com/sqs/v2/home?region=us-east-1#/queues

Queue wordfreq-results has been purged successfully.

Amazon SQS > Queues

Queues (2)

Search queues by prefix

Name	Type	Created	Messages available	Messages in flight
wordfreq-jobs	Standard	11/01/2022, 11:22:53 GMT	0	0
wordfreq-results	Standard	11/01/2022, 11:24:23 GMT	73	0

After the primary load testing with two different instance types, t1,t3 (except the primary instance used) , which had a time of around 15-20 mins depending on uploading time and processing time as shown below. I also setup a notification email service for changes in the autoscaling formation and termination to closely look at the costs that could be incurred.

The image displays four screenshots of the AWS Management Console interface, specifically focusing on Amazon SQS and S3 services.

**Top Screenshot (Jan 12 08:18):** Shows the Amazon SQS console with a table of queues. The table has columns for Name, Type, Created, Messages available, and Messages in flight.

Name	Type	Created	Messages available	Messages in flight
wordfreq-jobs	Standard	11/01/2022, 11:22:53 GMT	0	3
wordfreq-results	Standard	11/01/2022, 11:24:23 GMT	64	0

**Second Screenshot (Jan 12 08:07):** Similar to the first, but the 'Messages available' for 'wordfreq-jobs' is 0 and 'Messages in flight' is 0.

Name	Type	Created	Messages available	Messages in flight
wordfreq-jobs	Standard	11/01/2022, 11:22:53 GMT	0	0
wordfreq-results	Standard	11/01/2022, 11:24:23 GMT	0	0

**Third Screenshot (Jan 12 08:07):** Shows an S3 upload progress bar. The progress is at 0%. The text indicates: "Total remaining: 38 files: 145.8 MB(99.71%)", "Estimated time remaining: 44 minutes", and "Transfer rate: 57.1 KB/s".

**Bottom Screenshot (Jan 12 07:53):** Similar to the second screenshot, showing the Amazon SQS console with the same queue data.

Name	Type	Created	Messages available	Messages in flight
wordfreq-jobs	Standard	11/01/2022, 11:22:53 GMT	0	0
wordfreq-results	Standard	11/01/2022, 11:24:23 GMT	0	0

The autoscaling group was changed to a t3.micro with a different dynamic scaling setting, the cpu utilization requirements for new instances was reduced to 30% to decrease the time required to process files between instances and not trigger the instance timeout function in the sqs, leading to a rollback of the message request.

The image consists of four screenshots from the AWS Management Console, taken in a Brave Web Browser. The screenshots show the following:

- Top Screenshot (Jan 12 08:46):** The 'Amazon SQS' console page for the 'wordfreq' bucket. It shows two queues: 'wordfreq-jobs' and 'wordfreq-results'. The 'wordfreq-jobs' queue has 0 messages available and 0 messages in flight. The 'wordfreq-results' queue has 68 messages available and 0 messages in flight.
- Second Screenshot (Jan 12 08:44):** The same 'Amazon SQS' console page. The 'wordfreq-jobs' queue now has 4 messages in flight. The 'wordfreq-results' queue has 59 messages available and 0 messages in flight.
- Third Screenshot (Jan 12 08:35):** The same 'Amazon SQS' console page. The 'wordfreq-jobs' queue now has 28 messages available and 7 messages in flight. The 'wordfreq-results' queue has 7 messages available and 0 messages in flight.
- Bottom Screenshot (Jan 12 08:34):** The 'S3 Management Console' page showing the 'Upload: status' for the 'wordfreq' bucket. It displays a progress bar at 0% and indicates that 36 files (145.6 MB) are remaining to be uploaded, with an estimated time of 41 minutes and a transfer rate of 60.8 KB/s.



After the settings were changed, the time was reduced to half of the previous test. Most of it was due to the improvement in the instance type and reduction in the CPU utilization requirements that lead to higher allocation of instances and less rollback occurrences.

#### Company B:

Requirements: extremely cost-effective, efficient (scalable and resilient), occasional use, basic security and long-term data backups required.

#### Design –

Primary Instance – t2.micro, processes in a reasonable amount of time with low cost and enough memory to go through files in reasonable time.

Security group – Simple ssh rule to be accessed by the users for any changes and uploads, and https rule if the application is to be distributed to other employees. Proper creation of user groups with AWS IAM services to give proper rules and service access to avoid any mishaps.

AWS Cost Optimization – optimizes the cost of all services after analyzing the usage and suggests proper AWS services that can be used to maintain the performance but lower the costs.

DynamoDB- Simple NoSQL database, same as setup to store the results of the application

S3- Primary gateway to transfer uploaded files to job for applying the wordfreq application and can be auto scaled using lower values to create additional S3 in case of huge upload.

Glacier – archive storage provided by AWS can be used for long term backup or

AWS Backup – used for backing up anything on the cloud, can store the processed files that can be transferred to S3 bucket for processing again.

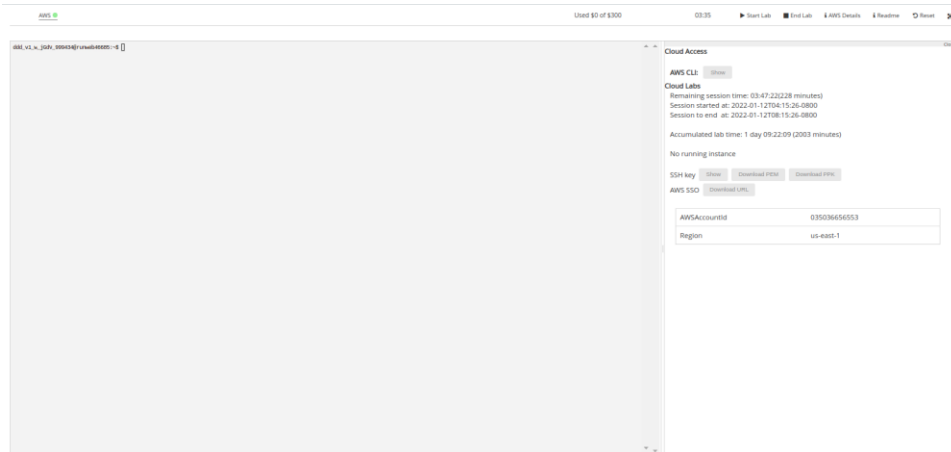
AWS compute optimizer – to optimize the compute services so we don't employ more resources than necessary given the situation.

AWS auto scaling for instances – helps in scaling in unseen circumstances but also provided proper resilience to the instances and perform health checks regularly to avoid any downtime of the application.

AWS Database migration service or AWS Snow – applicable to store the results in a physical or local field that can be kept on a physical storage which won't incur any cost at all.

Issues during the testing and working on wordfreq applications.

- No update to costs in the aws learner lab, even though it showed cost in the instances that would be incurred per hour, the update was not applied when starting or ending labs and as such I have not employed any heavy duty instances, to avoid any problems of running out of funds without realizing.



- Instance\_timeout due to network speeds and working but was managed by using a faster instance and better autoscaling optimization

```
an 12 08:19:58 ip-172-31-27-24 wordfreeservice[B74]: Deleted message, 7991f1f3-b922-4885-a484-d6d894922b98
an 12 08:19:58 ip-172-31-27-24 wordfreeservice[B74]: 2022/01/12 08:19:58 failed to process job 2179ae33-5865-4dd7-909d-2bea518e2fc Failed to update job messages's visibility timeout, InvalidParameterError: Value AQEB86-484vEge5n4L14uE3HsVhChnbl/IqInvlabeckIquLa+zrVnlyCZzVQf1p0Y9B53VVC2ACK/82p4f0s8pp/82p1k3u3V3CsDYp86nxbR8suV1CJ0nLA49xslwettfntf1b3hX/Bxand6q/8soxEHtCY2ERudkznJngud1212rBkXCznL8qg28d0lv8k02280y877ficekufB4kz2772f2xzy3f62AT2J81Pvz77Q7H13Q0hntLH5HAbCLQzszVP1jnvDUXJ02pY2u8p9w0GvMhygtp0akz1L8Nntf18J6anPKf15hntfPUNQ9PMLKt/RN0ZTgZyH0f2X68RoqSLCT0rerAYuy1lUAgpTE3q5D9prr83p9m= for parameter ReceiptHandle is invalid. Reason: Message does not exist or is not available for visibility timeout change.
an 12 08:19:58 ip-172-31-27-24 wordfreeservice[B74]: status code: 400, request id: aai40b7b-b35f-51e6-ab93-51232233db58
an 12 08:20:00 ip-172-31-27-24 wordfreeservice[B74]: Worker 0 received job 7991f1f3-b922-4885-a484-d6d894922b98
an 12 08:20:00 ip-172-31-27-24 wordfreeservice[B74]: Received job result 2179ae33-5865-4dd7-909d-2bea518e2fc
an 12 08:20:00 ip-172-31-27-24 wordfreeservice[B74]: 2022/01/12 08:20:00 Failed to process job 2179ae33-5865-4dd7-909d-2bea518e2fc
an 12 08:20:00 ip-172-31-27-24 wordfreeservice[B74]: 2022/01/12 08:20:00 failed to process job 7991f1f3-b922-4885-a484-d6d894922b98 Failed to update job messages's visibility timeout, InvalidParameterError: Value AQEBALvCY8d8t1D29p3114BHCaX3z0H8IT9MeLx895JMK1e5Uqy5J2Xo1fehve2Kup19wZn0b8ZUALTE3gUw421vGfHsdnTCMy8broZOG5vQ+LK+1ql09Jc3Gk+JfACP0omV2p0Y0H94wR3UZ0HPzY0M82Teuk8J)+8Zf73bheh0pfaN81Cay30n831bwech8k4tevp2Q1E1d0fF0c0x4e9fJdewtC0h0f17p0hId0of0W4fCq0kxAS4e2p0dZ11eP330P18h0T525w6d0mffJf0h0f0qum1+1b4y4gdcJog3Carnp0PAM0Xng0rr+kq1n1z1C3w0r1e87VcG5glQYgg= for parameter ReceiptHandle is invalid. Reason: Message does not exist or is not available for visibility timeout change.
```