

Table of Contents

Task 1: NHS and social care should use public cloud for data processing	2
Task 2: Scaling the WordFreq Application	4
Task A: Installing the application	4
Task B: Design and implement Auto-Scaling	5
Task C: Load Testing	10
Task D: Optimize the Word Freq Architecture	14
1. S3 Cross-Region Replication (CRR)	14
2. Amazon S3 Glacier	16
3. EC2 Spot Instance	17
4. Identity and Access Management (IAM)	19
Further Improvements	20

Table of Figures

Screen Shot 1 Launch Template Configuration	5
Screen Shot 2 Auto Scaling Configuration	6
Screen Shot 3 Number of Messages Received	7
Screen Shot 4: Scaling Out Alarm	8
Screen Shot 5: Scaling in Alarm	8
Screen Shot 6: Dynamic Policy	9
Screen Shot 7: Termination Policy	10
Screen Shot 8: SQS queue Message status	10
Screen Shot 9: Copied Files in Processing Bucket	11
Screen Shot 10: Instance Status in Auto Scaling Group	11
Screen Shot 11: EC2 Instance page	12
Screen Shot 12: Email received	12
Screen Shot 13: Optimized Cloud Architecture	14
Screen Shot 14: Create destination buckets	15
Screen Shot 15: Processing Bucket Replication Rule	16
Screen Shot 16: S3 Life Cycle Configuration	16
Screen Shot 17: Instance Type Requirement Configuration	17
Screen Shot 18: Instance Distribution	18
Screen Shot 19: Requesting Spot Instances	18
Screen Shot 20: Not Authorized to Perform	19
Screen Shot 21: Unauthorized to Perform Error	20

Task 1: NHS and social care should use public cloud for data processing

In recent years, there has been a strong interest within the health and social care system in the UK to adopt public cloud computing services. The use of these services has grown largely in many sectors and they have been proven to provide a cost-efficient and flexible way of managing and setting up data systems and architecture (Background and Context - NHS Digital, n.d.).

Amazon Web Service (AWS) is a cloud provider offering numerous cloud services that would benefit data processing for the NHS and Social Care sector. This essay aims to provide recommendations of AWS features and services that would be ideal and applicable for the requirements regarding security, availability, scalability, and cost optimization.

Amazon Elastic Cloud Computing (EC2) is a web service that provides scalable computing capacity on the cloud, EC2 instances come with different configurations by which healthcare organizations can choose and adjust their resource based on their demand for data processing workload. EC2 can integrate with other AWS services for storing, and managing databases such as S3 and RDS, this can simplify the workflow and ensure overall efficiency. Furthermore, EC2 follows a Pay-as-you-go pricing model which only charges for the computing capacity used, thus optimizing expenses for the organization (What Is Amazon EC2? - Amazon Elastic Compute Cloud, n.d.).

Simple Store Service (S3) is a service that offers scalable and secured object storage that can be suitable for healthcare data, such as patient records, medical images, etc. Through Amazon S3, healthcare organizations can access, manage, and archive their data securely across systems. In addition, Amazon S3 offers diverse storage classes that can serve varying purposes, organizations can decide which class they would prefer based on whether data must be frequently or infrequently accessed or to save cost by archiving the data with Amazon S3 Glacier. In terms of security, S3 configurations allow users to decide and manage whoever can have access to buckets and objects via Access Control Lists (ACLs) and Bucket Policies (What Is Amazon S3? - Amazon Simple Storage Service, n.d.).

DynamoDB stands out as a robust service that provides fast and flexible NoSQL database solutions and offers low latency access to data, which is especially crucial for the need for real-time data processing in healthcare sector (What Is Amazon DynamoDB? - Amazon DynamoDB, n.d.). DynamoDB operates as a fully managed service, which means the responsibility of set-up and

configuration tasks for the database will be relieved. Furthermore, sensitive data will also be protected with DynamoDB's encryption-at-rest feature.

The vast amount of health and social care data requires an equally robust and high-performance analytics tool. AWS addresses this need by offering Amazon Redshift which offers a simple and cost-efficient analytic solution. It is a powerful data warehousing service that enables users to carry out valuable analysis which helps to extract insight from patient records, disease trends, and patterns thus increasing productivity (Introduction - Amazon Redshift, n.d.). Additionally, users can have quick access to information thanks to Redshift's exceptional querying speed, which is beneficial in generating reports and conducting analysis in the NHS and social care sector.

In terms of Security and Access management, which is a primary concern for NHS and social care data, Amazon offers Identity and Access Management (IAM) services that enable organizations to have control over who can have access to their resource, databases, and services. Only users with permission granted through IAM roles and user groups can have access to the sensitive data and services across healthcare establishments and platforms (What Is IAM? - AWS Identity and Access Management, n.d.). Furthermore, an additional feature of IAM is Multi-factor authentication (MFA) allows you to specify two-factor authentication to the user account thus enhancing overall security.

Despite the benefits and efficiencies in operation that public cloud services would bring for data processing of NHS and social care, there are still circumstances where the use of public cloud services is deemed to be inappropriate. One of these scenarios concerns the location where the data will be hosted or processed (NHS And Social Care Data: Off-shoring and the Use of Public Cloud Services Guidance - NHS Digital, n.d.). On account of saving cost and increased availability, organizations would consider the multi-region strategy, having their data replicated across regions outside of the UK. However, there are some certain sensitive healthcare information that are deemed highly confidential and must comply with a country-specific data protection law. Furthermore, there exists a potential risk that overseas support staff of the cloud provider could have access to data that is protected by country-specific protection law (NHS And Social Care Data: Off-shoring and the Use of Public Cloud Services Guidance - NHS Digital, n.d.).

A second scenario would be when performance latency is a crucial requirement in the healthcare sector's practices. Specific practices and applications demand very low latency and process data

at high speed, such as real-time monitoring of patients which is vital in emergencies, in such cases the use of a public cloud would not be sufficient.

Lastly, when the network infrastructure of a healthcare facility is unable to meet the minimum network requirement for a stable implementation of the public cloud, it affects the cloud architecture's availability, causing unfavorable interruptions. Especially in the healthcare sector, access to data be queried seamlessly, therefore, when the network requirement is insufficient, data availability cannot be guaranteed, and the use of public clouds should be avoided.

In conclusion, the health and social care system in the UK is showing strong interest in the integration of public cloud services in data processing, AWS stands as a perfect cloud provider as its solutions provide security, availability, scalability, and cost optimization for its users. Features and services such as Amazon EC2, Amazon S3, DynamoDB, Amazon RedShift, and IAM are recommended to be ideal for the healthcare sector's data management. However, before implementing public clouds, consideration regarding regulation, requirements of minimal latency, and network infrastructures must be taken into account.

Task 2: Scaling the WordFreq Application

Task A: Installing the application

Word Freq is an application used for counting the top 10 most frequently found words in a text file. The functionality of the application can be briefly described as follows.

- Text files are saved from the local machine or uploaded by users and stored in the S3 Uploading Bucket. Once the files are copied to the Processing Bucket, a message containing a text file will be added to the SQS queue wordfreq-jobs and simultaneously a user will receive an email through SNS.
- The message is now queued in the wordfreq-jobs SQS queue, the application detects the jobs and triggers the word-count process. The results generated are then saved in a DynamoDB table and the results processed are contained in a message and added to the wordfreq-result queue.
- The application is hosted and managed on an EC2 instance, connecting and interacting with the S3 Bucket, SQS queue, DynamoDB, and the overall process flow.

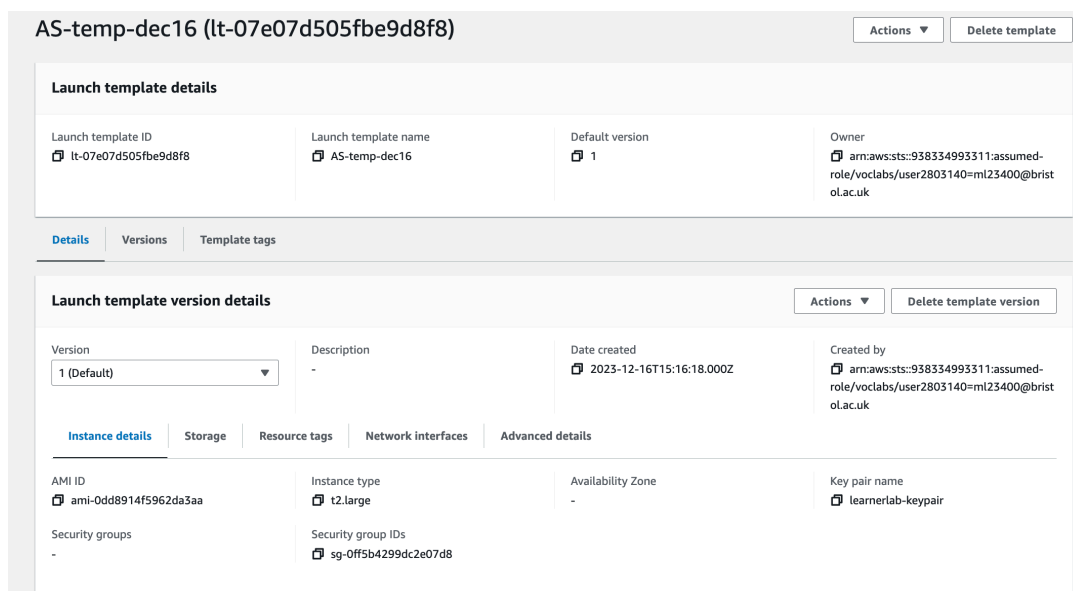
The application is functional but numerous improvements can still be made. Additional AWS services and features can be integrated to fulfill the limitations of the application concerning availability, resilience against failure, data protection and security, and cost optimization.

Task B: Design and implement Auto-Scaling

Before applying Auto-Scaling to the application, the recorded time that the Word Freq application without the auto-scaling stands at 27 minutes.

Step 1: Create a launch template

Launch templates allow you to save instance configuration for auto-scaling activity, and it can have multiple versions for each instance type and configuration. The following screenshot gives information on the configurations of the first version of the Launch template.



Screen Shot 1 Launch Template Configuration

Step 2: Create an Auto-Scaling group

After setting up the launch template, we will move on to setting up the Auto-Scaling group, the following screenshot gives information on the configuration of the Auto-Scaling group, here the maximum capacity is defined at 4 while the minimum and desired capacity of instance is defined

as 1, as there must always be at least one worker instance available for processing messages.

AS-Group

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Group details Edit

Auto Scaling group name AS-Group	Desired capacity 1	Desired capacity type Units (number of instances)	Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:938334993311:autoScalingGroup:eb4f8040-02be-4243-aeb2-4e5183095c4f:autoScalingGroupName/AS-Group
Date created Sat Dec 16 2023 15:18:36 GMT+0000 (Greenwich Mean Time)	Minimum capacity 1	Status <input checked="" type="radio"/> Updating capacity	
	Maximum capacity 4		

Launch template Edit

Launch template lt-07e07d505f9e9d8f8 AS-temp-dec16	AMI ID ami-0dd8914f5962da3aa	Instance type t2.medium	Owner arn:aws:sts:938334993311:assumed-role/voclabs/user2803140=ml23400@bristol.ac.uk
Version 7	Security groups -	Security group IDs sg-0ff5b4299dc2e07d8	Create time Thu Dec 21 2023 14:35:01 GMT+0000 (Greenwich Mean Time)
Description ver-t2medium	Storage (volumes) -	Key pair name learnerlab-keypair	Request Spot Instances No

[View details in the launch template console](#)

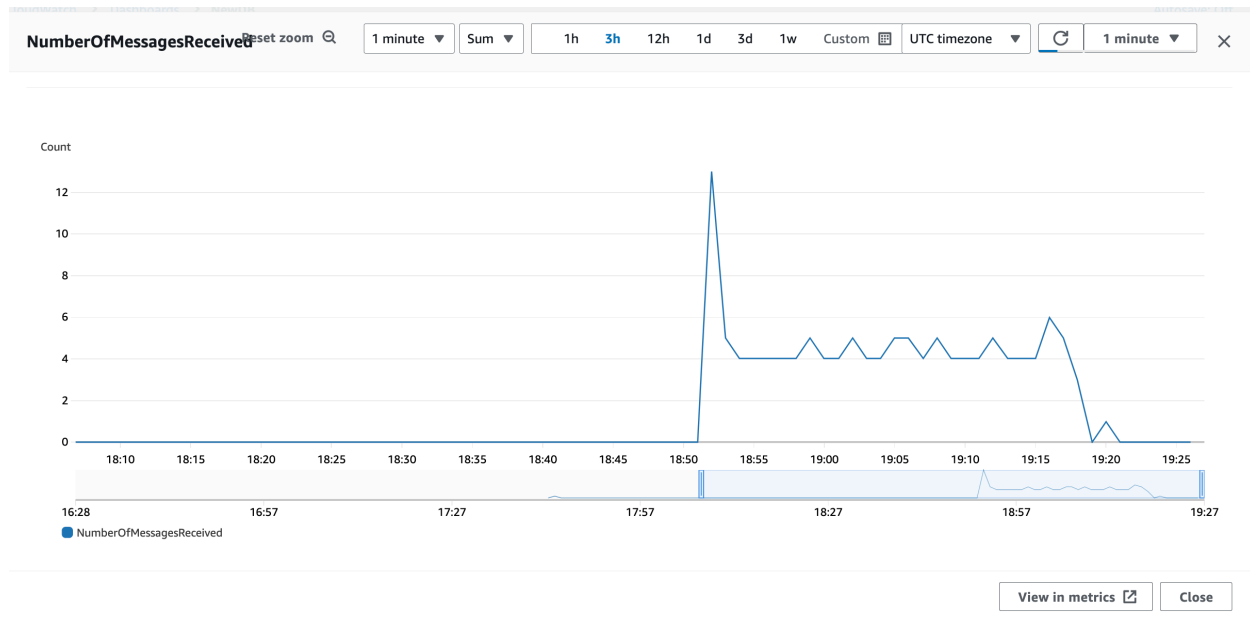
Screen Shot 2 Auto Scaling Configuration

Step 3: Selecting metrics and creating CloudWatch Alarm

Word Freq application function is based on the transferring of messages containing text files in the SQS queue from the S3 uploading bucket and waiting to be processed by Worker Application, therefore the most suitable metric would be associated with the queue metric of SQS.

After looking at the shape of the metric graphed by CloudWatch the ideal option for auto-scaling action of the Word Freq application based on its functionality would be **“NumberOfMessagesReceived”** for SQS queue wordfreq-jobs, as illustrated in the graph below, the metric can be seen as having high granularity, indicating that the metric is sensitive to application’s response time and fluctuation of performance, which could enable Auto Scaling to react quickly to changes in the number of workloads, thus triggering the autoscaling action promptly.

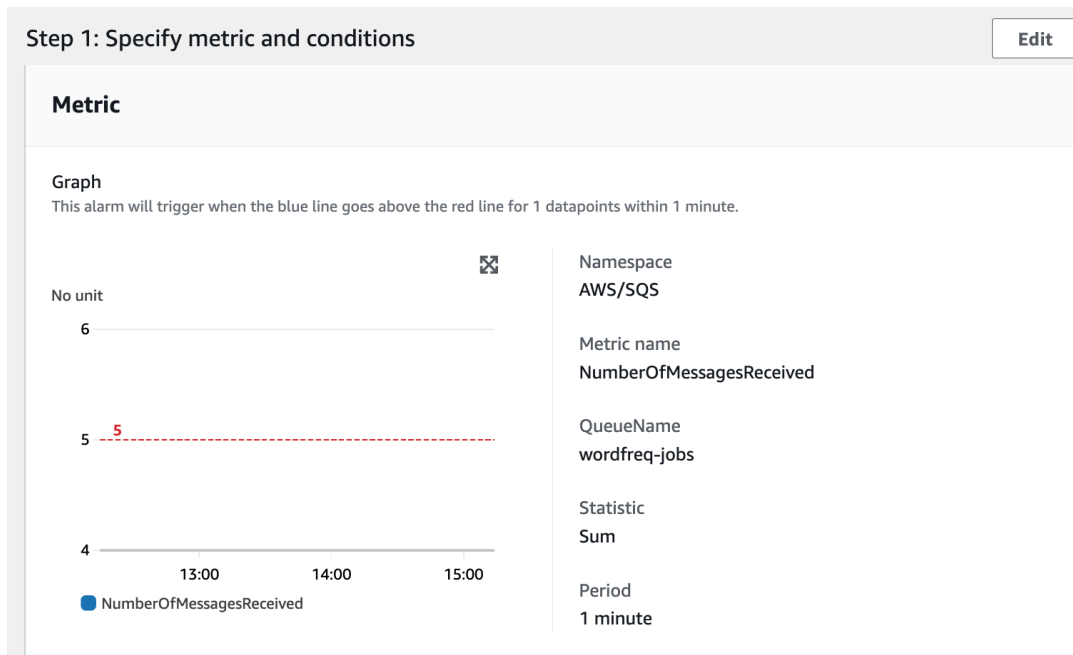
The graph depicts the metric’s appearance before the application of auto-scaling



Screen Shot 3 Number of Messages Received

After selecting our desired metric, we'll now move on to creating alarms for the lower and upper thresholds.

In the load test of auto-scaling, the upper and lower thresholds for scaling out and scaling in the Cloud Watch alarm will be defined at 5 and 4 respectively. The screenshots below show the configurations and for the "Scale in" and "Scale out" alarms, the statistic is set at "Sum" and Period at "1 minute".



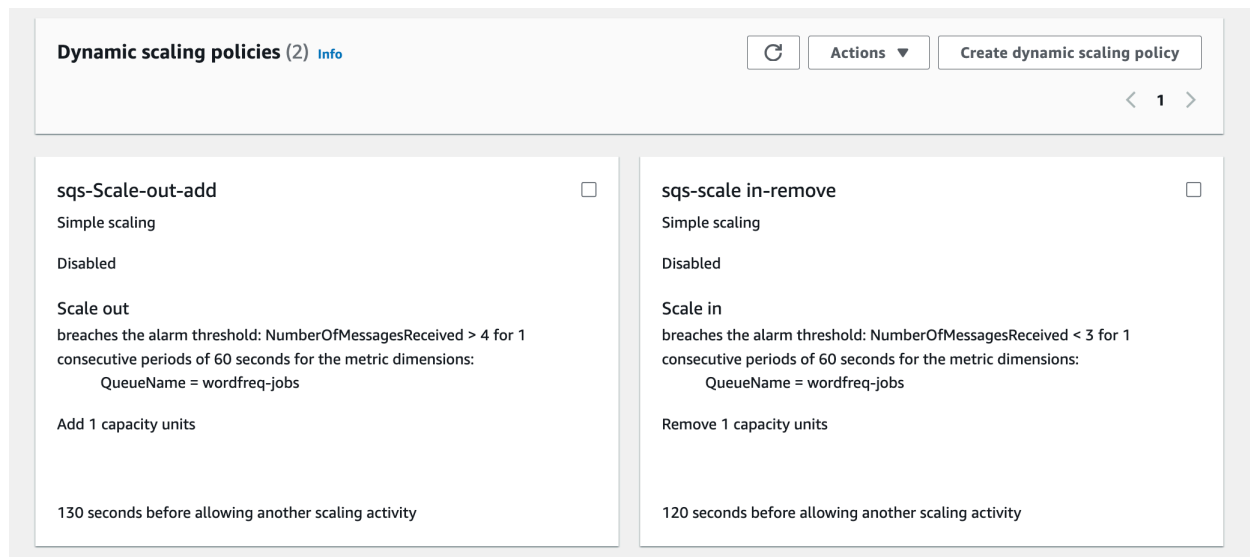
Screen Shot 4: Scaling Out Alarm



Screen Shot 5: Scaling in Alarm

Step 4: Create a Dynamic Scaling policy

After Cloud Watch alarms have been established and operate normally, we can start creating a Dynamic Scaling Policy and specify when Auto-Scaling action can be triggered based on Cloud Watch Alarm, we will create 2 dynamic scaling policies, one is used for scaling out and one for scaling in.



Screen Shot 6: Dynamic Policy

For the scaling out policy, 130 seconds of cool down time before adding another instance was applied since it is required that only one instance can be added every 2 minutes. In the first load test, the cool-down duration is established at 120 seconds.

Note: In Auto Scaling Group, the termination policy needs to be configured at the “Newest Instance”. This configuration makes sure that when the alarm for scaling-in metric threshold is triggered, Auto Scaling will terminate the most recently created.

Edit AS-Group [Info](#)

Advanced configurations

Instance scale-in protection [Info](#)

If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ Enable instance scale-in protection

Termination policies [Info](#)

The termination policies used to select the instance to terminate during scale in, listed in priority order from highest to lowest.

Order Termination policies (Drag and drop policies to change their order)

1



Newest Instance



Add policy

Screen Shot 7: Termination Policy

Task C: Load Testing

In the first test, the instance type in the Launch Template version specified as t2.micro, and the upper and lower threshold for CloudWatch Metric are 4 and 3 respectively.

- Message status in SQS queue while running application

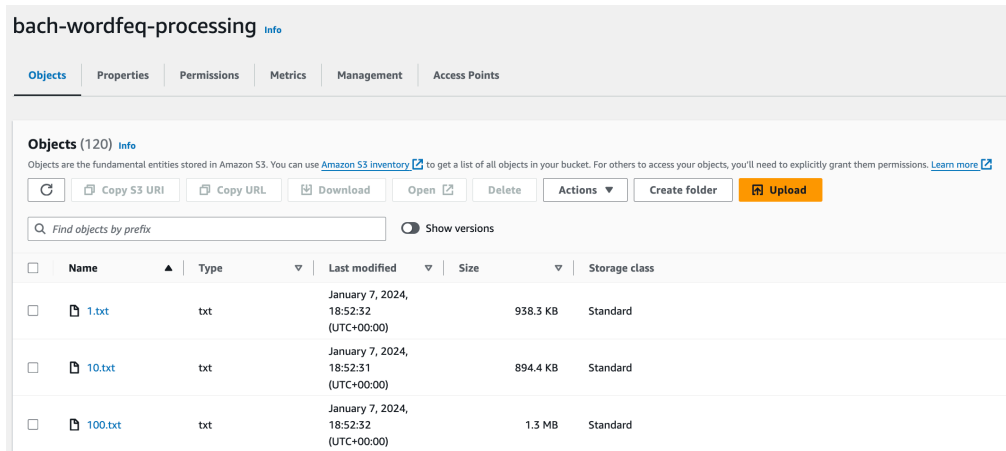
Amazon SQS > Queues

Queues (2) [Refresh queues table](#) [Edit](#) [Delete](#) [Send and receive messages](#) [Actions](#) [Create queue](#)

	Name ▲	Type ▼	Created ▼	Messages available ▼	Messages in flight ▼	Encryption
<input type="radio"/>	wordfreq-jobs	Standard	2023-12-03T15:45+00:00	14	36	Amazon SQS key (SS
<input type="radio"/>	wordfreq-results	Standard	2023-12-03T15:47+00:00	63	0	Amazon SQS key (SS

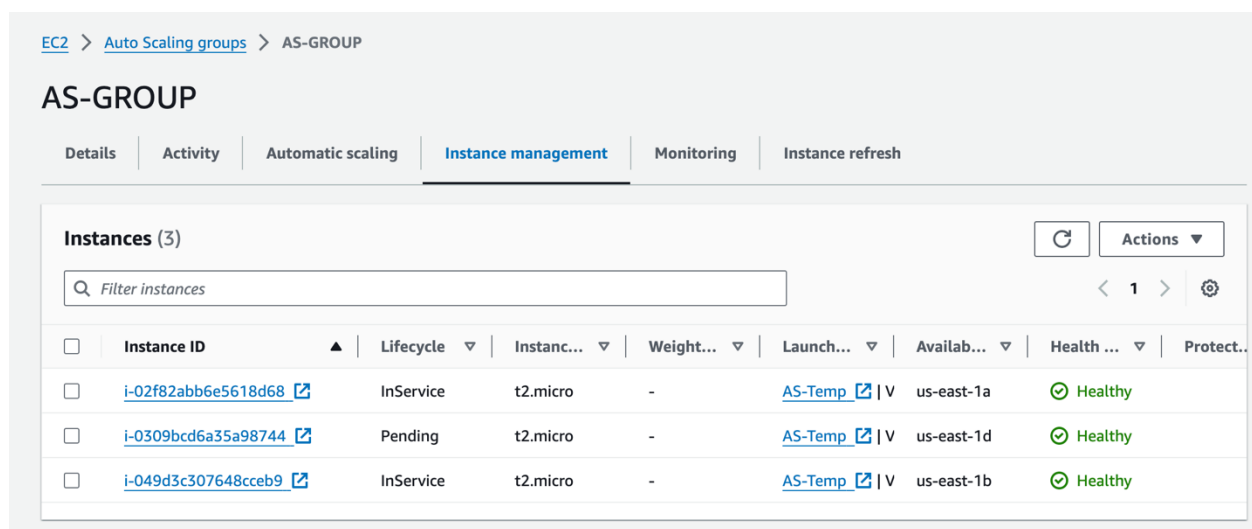
Screen Shot 8: SQS queue Message status

- 120 files copied in Processing Bucket



Screen Shot 9: Copied Files in Processing Bucket

- Auto Scaling Group page showing instance status



Screen Shot 10: Instance Status in Auto Scaling Group

- EC2 instance page showing launched/terminated instances

Instances (5) Info

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Find Instance by attribute or tag (case-sensitive)

<

1

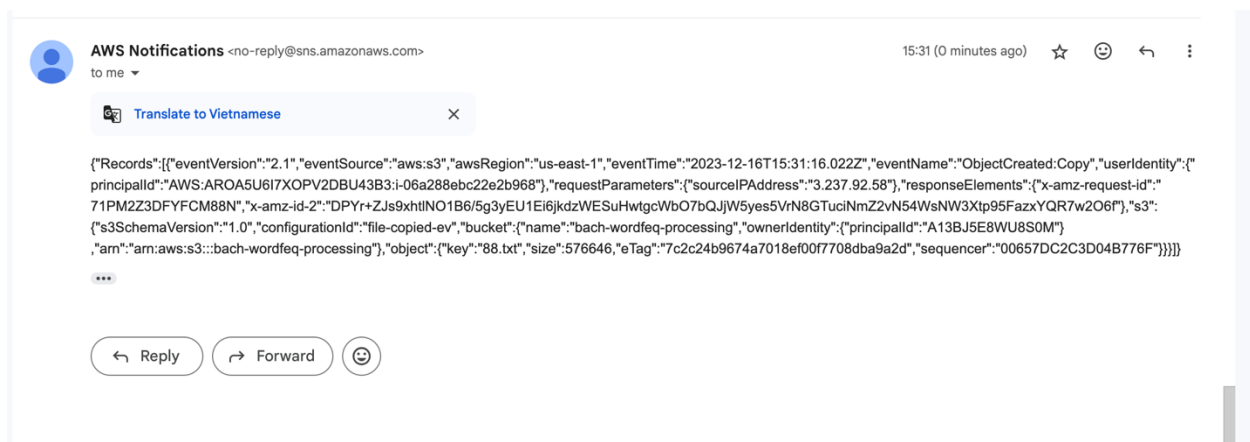
>

{

<input type="checkbox"/>	Name <div></div>	Instance ID	Instance state <div></div>	Instance type <div></div>	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	as-activity-inst...	i-049d3c307648cceb9	<div></div> Running <div></div>	t2.micro	<div></div> 2/2 checks passed	View alarms +	us-east-1b
<input type="checkbox"/>	as-activity-inst...	i-02f82abb6e5618d68	<div></div> Running <div></div>	t2.micro	<div></div> 2/2 checks passed	View alarms +	us-east-1a
<input type="checkbox"/>	as-activity-inst...	i-0309bcd6a35a98744	<div></div> Terminated <div></div>	t2.micro	-	View alarms +	us-east-1d
<input type="checkbox"/>	wordfreq-dev	i-015dd54ccd26340fd	<div></div> Stopped <div></div>	t2.micro	-	View alarms +	us-east-1b
<input type="checkbox"/>	as-activity-inst...	i-01e7a421903e84961	<div></div> Terminated <div></div>	t2.micro	-	View alarms +	us-east-1c

Screen Shot 11: EC2 Instance page

- Emails received from Amazon S3 Notification



Screen Shot 12: Email received

In this section of the report, multiple load tests will be performed with different configurations and instance types, this can be done by creating different versions of Launch Template corresponding with each type of instance's CPU power, memory, etc. The table below describes the result of each load test experiment and their corresponding processing time and estimated cost:

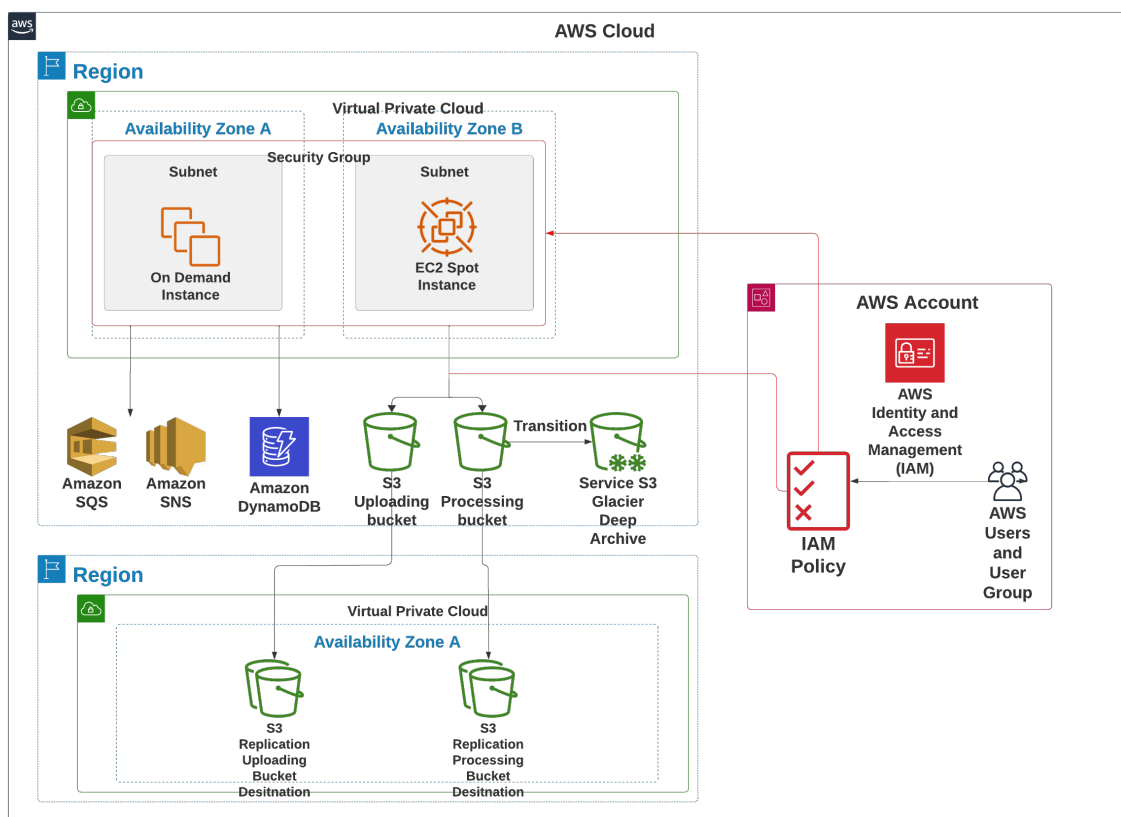
Table 1: Load Test Results

Trial	Instance created	Upper threshold	Lower threshold	Instance Type	Processing time (minute)	Wait time for terminating an instance (second)	Cost (\$)
0	-	-	-	t2.micro	27	-	0.0052
1	4	5	4	t2.micro	12	120	0.0096
2	4	4	3	t2.micro	10	120	0.0085
3	4	4	3	t2.micro	11	60	0.0079
4	4	4	3	t2.micro	11	30	0.0072
5	4	4	3	t2.small	11	30	0.0180
6	3	4	3	t2.medium	7	30	0.0162
7	3	4	3	t2.large	6	30	0.0147
8	3	4	3	t3.micro	8	30	0.0043
9	3	4	3	t3.small	7	30	0.0076
10	3	4	3	t3.medium	7	30	0.0166
11	3	4	3	t3.large	6	30	0.0332

Based on the results of the load tests, it can be seen that the upper and lower threshold configuration which gives the best performance are 4 and 3 respectively. Moreover, defining the waiting period for terminating an instance at 30 seconds is the most cost-effective configuration as instances are terminated swiftly when not being required. It can be seen that out of 11 load tests, EC2 instance t2.large offers the best performance with the shortest processing time at six minutes and costs only \$0.0147. In comparison with the EC2 instance t3.large process in six minutes but cost \$0.0332, which is notably higher.

Task D: Optimize the Word Freq Architecture

This section of the report presents the redesigned cloud architecture for the Word Freq application using services and features introduced in the Cloud Foundation course. The updated architecture will include the integration of Amazon S3 Cross Regional Replication and Glacier to improve resilience, availability, and data backup. IAM is utilized to mitigate the concern of security and unauthorized access to resources and applications. For cost optimization and to meet the application's occasional usage demand, EC2 Spot Instance will be integrated as an additional cost-effective service option. The diagram below describes the optimized cloud architecture.



Screen Shot 13: Optimized Cloud Architecture

1. S3 Cross-Region Replication (CRR)

S3 Cross Region Replication is used to automatically and asynchronously create copies of objects across Amazon S3 buckets in different AWS regions, it ensures that data are backed up, in case of issue or disaster in the original region where data is stored, and valuable data can still be

recoverable from other regions, thus minimizing interruption in business workflow and the risk of data loss (Replicating Objects - Amazon Simple Storage Service, n.d.).

Users in different regions can benefit from reduced latency as they can access data from regions close to them, improving performance, this is especially useful for services and applications that are deployed globally or must be accessed remotely, enhancing high availability.

The first step to applying S3 Cross-Region Replication to our Word Freq application would be creating a destination bucket in a different region, two buckets created are “processing-destination” and “uploading-destination”. The screenshot shows that the destination bucket is stored in different regions compared to the main bucket.

General purpose buckets (5) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#) 

 Find buckets by name

	Name	AWS Region
<input type="radio"/>	aws-logs-938334993311-us-east-1	US East (N. Virginia) us-east-1
<input type="radio"/>	bach-wordfreq-processing	US East (N. Virginia) us-east-1
<input type="radio"/>	bach-wordfreq-uploading	US East (N. Virginia) us-east-1
<input type="radio"/>	processing-destination	US West (Oregon) us-west-2
<input type="radio"/>	uploading-destination	US West (Oregon) us-west-2

Screen Shot 14: Create destination buckets

The next step is to create a Replication Rule in the Management tab of the chosen bucket. When creating a replication rule Bucket Versioning needs to be enabled and we can specify a storage class for the replicated object. For the Word Freq application, we will set the storage class at Standard-IA (for infrequently access data with millisecond access) to save cost as the destination bucket will be mostly served for data recovery and therefore will not be accessed frequently.

Below are screenshots of the process bucket’s replication configurations, the storage class is defined as “Transition to Standard-IA”.

Replication rules (1)
Use replication rules to define options you want Amazon S3 to apply during replication such as server-side encryption, replica ownership, transitioning replicas to another storage class, and more. [Learn more](#)

Replication rule name	Status	Destination bucket	Destination Region	Priority	Scope	Storage class	Replica owner	Replication Time Control	KMS-encrypted objects (SSE-KMS or DSSE-KMS)	Replica modification sync
<input type="radio"/> default-rule	Enabled	s3://processing-destination	US West (Oregon) us-west-2	0	Entire bucket	Transition to Standard-IA	Same as source	Disabled	Do not replicate	Disabled

Screen Shot 15: Processing Bucket Replication Rule

2. Amazon S3 Glacier

S3 Glacier is a cost-efficient, long-term storage service that ensures security and durability for your infrequently accessed data. In S3 Glacier, data are backed up and archived using vaults. Objects stored in normal S3 buckets can be transitioned to S3 glacier for archiving using Amazon S3 Lifecycle, by defining transition rules, objects will be transitioned into another storage class based on how frequently they are accessed throughout their lifecycle, thus saving cost of storage and making sure that data are archived and stored securely (What Is Amazon S3 Glacier? - Amazon S3 Glacier, n.d.).

To apply Amazon S3 Glacier into our architecture, first, we need to create a life cycle rule for our uploading and processing buckets, The life cycle rule can be found in the Management tab of the bucket. In the life cycle creation page, we can specify the number of days after object creation that object class is transitioned into another class. From the screenshot, it can be seen that after 6 months, data will be transitioned to the Glacier Instant Retrieval class and after 1 year, they will be moved into Glacier Deep Archive, maintaining a low-cost storage of infrequently accessed data.

Transition current versions of objects between storage classes
Choose transitions to move current versions of objects between storage classes based on your use case scenario and performance access requirements. These transitions start from when the objects are created and are consecutively applied. [Learn more](#)

Choose storage class transitions	Days after object creation	
Standard-IA	30	<input type="button" value="Remove"/>
Glacier Instant Retrieval	180	<input type="button" value="Remove"/>
Glacier Deep Archive	365	<input type="button" value="Remove"/>

Screen Shot 16: S3 Life Cycle Configuration

3. EC2 Spot Instance

EC2 Spot Instance is an instance option that utilizes unused EC2 capacity, making their price substantially less than the On-demand price plan which can help you to substantially optimize the cost of Amazon EC2 (Spot Instances - Amazon Elastic Compute Cloud, n.d.). EC2 Spot instances are exceptionally well-suited to applications with flexible processing demand, and dynamic workload and can tolerate interruption.

To integrate EC2 Spot Instance into our Word Freq application, the best practice would be to combine the use of Spot Instance and On-demand EC2 through Auto Scaling. Firstly, we must create a new Auto Scaling group called “Mixed”. Proceed to step 2 “Choose instance launch options” We then select option “Override Launch Template” to specify instance attribute and distribution of On-Demand and Spot instances.

Here the minimum vCPU and Memory (Gib) are set to 1 according to the configuration of t2.medium:

Instance type requirements [Info](#)

Reset to launch template

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

☒ **Specify instance attributes**
Provide your compute requirements. We fulfill your desired capacity with matching instance types based on your allocation strategy selection.

☐ **Manually add instance types**
Add one or more instance types. Any of the instance types may be launched to fulfill your desired capacity based on your allocation strategy selection.

Required instance attributes
Enter your compute requirements in virtual CPUs (vCPUs) and memory.

vCPUs
Enter the minimum and maximum number of vCPUs per instance.

minimum maximum

☐ No minimum ☒ No maximum

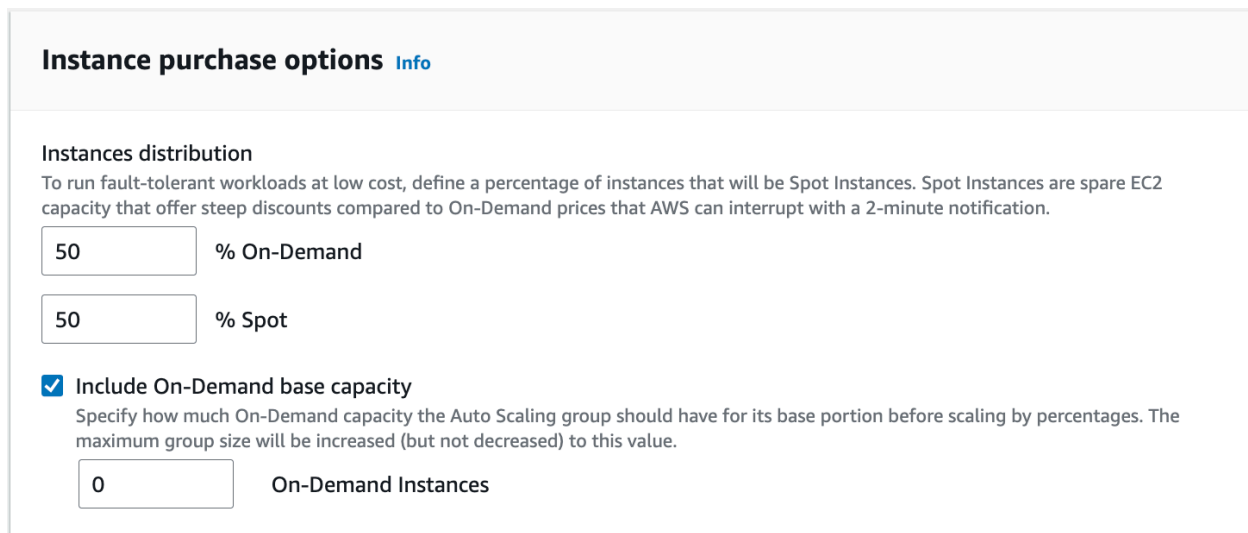
Memory (GiB)
Enter the minimum and maximum GiBs of memory per instance.

minimum maximum

☐ No minimum ☒ No maximum

Screen Shot 17: Instance Type Requirement Configuration

The instance purchase option setting is set to 50% of On-demand instances and 50% Spot instances to ensure workflow in case of Spot instance interruption.



Instance purchase options [Info](#)

Instances distribution
To run fault-tolerant workloads at low cost, define a percentage of instances that will be Spot Instances. Spot Instances are spare EC2 capacity that offer steep discounts compared to On-Demand prices that AWS can interrupt with a 2-minute notification.

% On-Demand

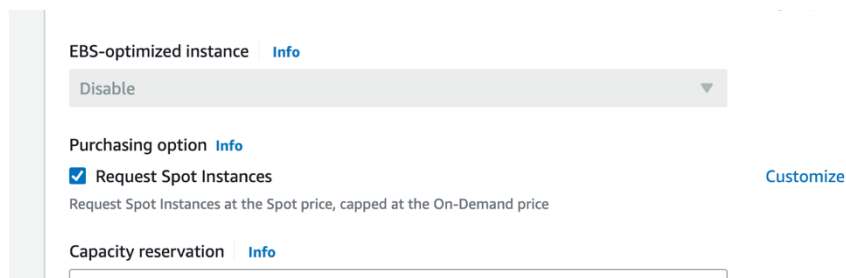
% Spot

☒ **Include On-Demand base capacity**
Specify how much On-Demand capacity the Auto Scaling group should have for its base portion before scaling by percentages. The maximum group size will be increased (but not decreased) to this value.

On-Demand Instances

Screen Shot 18: Instance Distribution

However, when attempting to request spot instances in the Launch Template Page, I encountered an error stating that the operation is not authorized to perform. Therefore, I could not implement EC2 Spot Instance into my Architecture.



EBS-optimized instance [Info](#)

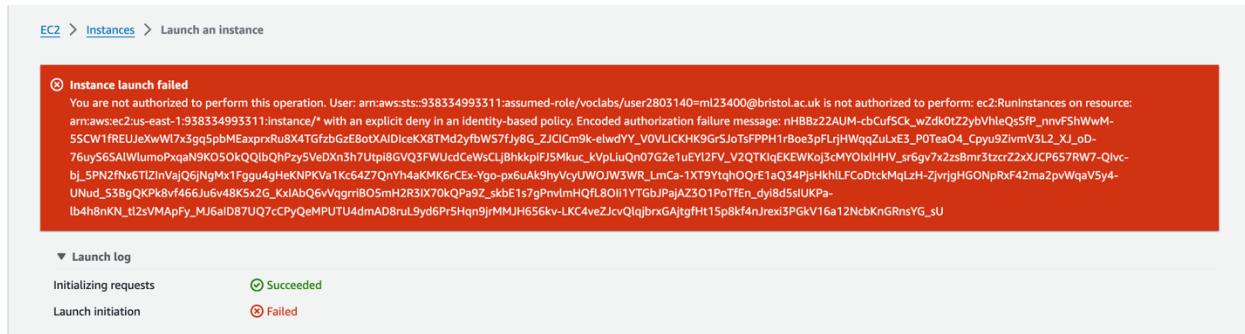
Disable ▼

Purchasing option [Info](#)

☒ **Request Spot Instances** [Customize](#)
Request Spot Instances at the Spot price, capped at the On-Demand price

Capacity reservation [Info](#)

Screen Shot 19: Requesting Spot Instances



Screen Shot 20: Not Authorized to Perform

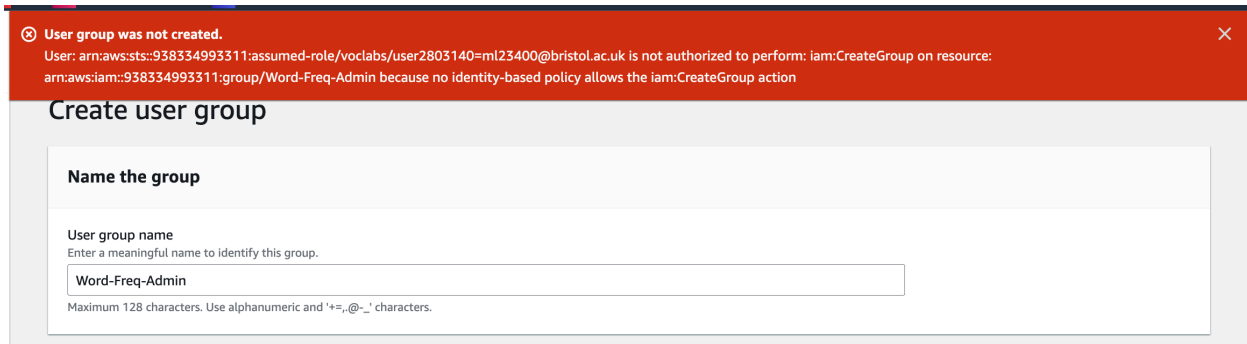
4. Identity and Access Management (IAM)

To prevent unauthorized access to the application data and resources, this report suggests implementing AWS Identity and Access Management (IAM). AWS IAM allows you to decide which user has permission to access the application resource, which resource can they have access to, and to what extent the user is allowed to handle the resource or data (What Is IAM? - AWS Identity and Access Management, n.d.). AWS IAM also does not incur any fee toward its user, it is a no-cost feature.

Specifically for the Word Freq application, we can grant permission to different members of the development team to have access to certain features and resources, such as only a limited number of users from a user group can have permission to have full access to Amazon EC2, Amazon S3, Amazon DynamoDB...

To implement IAM into the Word Freq application we need to specify Users in to User Group with a predefined Permission Policy. One user group named Word Freq-Admin will have most access would include the following Permission Policy: *AmazonEC2FullAccess*, *AmazonS3FullAccess*, *AmazonDynamoDBFullAccess*, *AmazonSQSFullAccess*, *AmazonSNSFullAccess*. Once users are added to the group, only Admins will have full access to all of the application resources.

Upon attempting to create a User Group, I encountered an error as the account is not permitted to create a user group:



Screen Shot 21: Unauthorized to Perform Error

Further Improvements

For a more robust and high-performance data processing application introduced in the LSDE course, two alternative applications that would be well-suited for the Word Freq application are Apache Hadoop and Apache Spark. Both are large-scale data technologies and can be used to manage and process large volumes of data, which makes them highly suitable for a word-counting application that requires processing capability of an amount of data, resilience, and fault tolerance. Hadoop is a noteworthy alternative to AWS as it can process and store large amounts of data quickly and efficiently (Simplilearn, 2020). With the Hadoop Distributed File System, data is stored and processed in multiple systems instead of one central system across the Hadoop cluster, thus enabling scalability, furthermore, data can be replicated across many nodes which reduces the risk of data loss thus enhancing fault tolerance. At the same time, Hadoop ensures data integrity and security while providing end-to-end encryption at rest and in transit. The MapReduce framework allows Hadoop to perform parallel processing of large volumes of data on different slave nodes.

With Apache Spark, data are stored in RAM, enabling swift access for data retrieval and rapid data analytics, therefore Spark has a remarkable processing speed as it can run up to 10 to 100 times faster in comparison with Hadoop (Simplilearn, 2019). Spark's compatibility with multiple languages makes it easier for developers to use, interact, and write applications on Spark. To fulfill the demanding fault tolerance requirement of numerous applications and developers, Spark introduced Resilient Distributed Datasets (RDDs) that are designed to handle failures from worker nodes, the approach will significantly minimize the risk of data loss.

REFERENCE:

Background and context - NHS Digital. (n.d.). NHS Digital. <https://digital.nhs.uk/data-and-information/looking-after-information/data-security-and-information-governance/nhs-and-social-care-data-off-shoring-and-the-use-of-public-cloud-services/cloud-risk-framework/background-and-context>

Introduction - Amazon Redshift. (n.d.).

<https://docs.aws.amazon.com/redshift/latest/dg/welcome.html>

NHS and social care data: off-shoring and the use of public cloud services Guidance - NHS Digital. (n.d.). NHS Digital. <https://digital.nhs.uk/data-and-information/looking-after-information/data-security-and-information-governance/nhs-and-social-care-data-off-shoring-and-the-use-of-public-cloud-services/guidance#benefits-of-the-cloud>

Replicating objects - Amazon Simple Storage Service. (n.d.).

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/replication.html>

Simplilearn. (2019, December 6). *Hadoop vs Spark | Hadoop And Spark Difference | Hadoop And Spark Training | Simplilearn* [Video]. YouTube.

<https://www.youtube.com/watch?v=2PVzOHA3ktE>

Simplilearn. (2020, November 29). *Introduction to Hadoop | Hadoop explained | Hadoop Tutorial for Beginners | Hadoop | Simplilearn* [Video]. YouTube.

<https://www.youtube.com/watch?v=hLnB0uzGvDI>

Spot instances - Amazon Elastic Compute Cloud. (n.d.).

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances.html>

What is Amazon DynamoDB? - Amazon DynamoDB. (n.d.).

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>

What is Amazon EC2? - Amazon Elastic Compute Cloud. (n.d.).

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

What is Amazon S3? - Amazon Simple Storage Service. (n.d.).

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>

What is Amazon S3 Glacier? - Amazon S3 Glacier. (n.d.).

<https://docs.aws.amazon.com/amazonglacier/latest/dev/introduction.html>

What is IAM? - AWS Identity and Access Management. (n.d.).

<https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html>

AWS Academy Account Credential

Username: ml23400@bristol.ac.uk

Password: Iamir0nman?