github\_pat\_11A3NC3TA0WyeC0XqSj7HJ\_IiyOidB20bq0L9bFbo7GW2W0QBvsCoSeMWv221TU0a7UCSOX3R48guYlcS2

**Task 1**

Cloud Computing has become increasingly popular in recent times with many companies using the public cloud such as AWS for their services that can help with scalability, flexibility and ultimately improve their day-to-day operations. Many financial institutions have integrated the public cloud for data processing, and I will discuss whether they should use it.

Financial Institutions have a lot to gain when using a public cloud, especially since AWS have started to focus on improving financial applications since 2017 and now over 90% of financial institutions have implemented some form of cloud computing in their infrastructure. One of the main advantages of the public cloud over using other resources for data is that it is highly dynamic, scalable, reliable and use the pay as you go pricing model. AWS offer many resources that can benefit financial institutions for data processing and one of the main resources they offer are the compute services. AWS offer many different types of compute services but the main 3 are Amazon Elastic Compute Cloud (EC2), Amazon Lambda and Amazon Elastic Beanstalk. Amazon EC2 is the most beneficial for financial institutions as it uses the Infrastructure as a Service (IaaS) cloud structure meaning that it gives full control over the maintenance of the software whilst not having to manage the underlying infrastructure. EC2 is an instance-based service that uses virtual machines that you can provision and manage. One of biggest advantages of EC2 is that there are several instance types that the financial institution can choose from based on their needs such as c4 and c5 for compute optimisation that provide high performance. There are also many storage options with EC2 such as Elastic Block Store (EBS) that maintains the data even after stopping the instance. Encryption can also be used. A case study found that Bankinter made use of Amazon EC2 by running over 5,000,000 simulations through a grid of multiple Amazon EC2 instances. This helped reduce their compute time from 23 hours to 20 minutes which improved the efficiency of the bank through time savings, along with cost savings. A quote from Javier Roldan, the Director of Technological Innovation “The AWS platform, with its unlimited and flexible computational power, is a good fit for our risk-simulation process requirements” illustrates that the use of the public cloud has been a benefit for this financial institution. (no.1)

Not only can financial institutions utilise the public cloud for compute power, but public clouds also such as AWS offer other services such as Databases. Financial Institutions are required to hold vast amounts of data such as financial transaction history. The key advantage of using a public cloud for databases is that AWS use managed database services which allows the financial institution to leave AWS to provision and manage the database itself and therefore requires less resources to be used for the company. Financial Institutions usually require relational databases (MySQL) due to ACID. Many FI’s have started to integrate more Internet of Things and AI apps that also require NoSQL, non-relational databases to access the data faster and more cheaply than you can with a MySQL database. AWS offers Amazon Relational Database Service (RDS) that allows you to provision numerous relational database systems such as MySQL and Oracle. The benefit of this is that you can choose a multiple availability zone (AZ) deployment along with automatic EBS snapshots that are taken every 5 minutes to ensure high availability (roughly 99.95%) and to allow for disaster recovery. DynamoDB would be very useful for financial institutions developing new apps as it is a fully managed, serverless non-relational database service that can run apps at any scale. The advantage of using DynamoDB over a non-cloud database is that it offers built-in security, it can scale horizontally achieving low latency read and writes for an infinite number of items in the table, along with consistent performance during peak times. (Possibly put some information about Amazon Redshift too as this is good for financial,sales and inventory – can then generate reports too)

In my opinion, the most valuable feature of using a public cloud is its ability to auto-scale on demand. Auto-scaling will allow financial institutions to scale their applications to ensure maximum availability and to provision only the capacity that is required to save resources and money. Amazon CloudWatch is a service that allows you to monitor your resources and applications that run on AWS, and you can configure alarms that trigger based off your specifications and thresholds you apply. This is beneficial to financial institutions as they require high uptime of their applications. Auto-scaling can also be used for EC2 where you can scale your application through using auto scaling groups that can be configured to scale in or out based on current circumstances. Dynamic scaling would be very useful for financial institutions as their traffic and workloads change all the time. By implementing dynamic scaling where an elastic load balancer is used to distribute traffic to different EC2 instances in multiple AZs, it not only allows for maximum availability, but it drastically reduces costs as only the requires amount of EC2 instances are used. Predictive scaling is also available but requires atleast 24 hours of historical data to start forecasting.

A case study found that finflux, an Indian financial institution reduced their IT costs by around 40% by using AWS Auto-Scaling. This case study is a great example of why financial institutions should use public cloud for data processing because before using AWS, the cost of scaling of the on-premise infrastructure to handle micro-loans outweighed the returns they actually received from the loans. (no.2)

\*include transfer ability in the cloud\*

On the other hand, the public cloud should be avoided for financial institutions for data processing mainly for security reasons. Financial institutions deal with large amounts of sensitive data such as customers bank details and transaction records. Financial Institutions specifically are heavily targeted by hackers through the nature of their business and are therefore highly regulated by regulators such as the British Financial Conduct Authority (FCA). I think that all of the sensitive data that financial institutions hold should be not stored on the public cloud and should be either kept on-premises or stored in a private cloud so that not only is the sensitive data more secure, it allows them to comply with regulations easier.

Another scenario is that generally financial institutions prefer using the capital expenditure model (CapEx) over the operating expenditure model (OpEx). As the public cloud adopts the pay as you go model, it primarily uses the OpEx model meaning that some financial institutions are weary of using it.

Finally, a scenario where a financial institution should not use the public cloud for data processing may not be a technological one, it may be because they risk changing their entire landscape and infrastructure that they have built up for multiple years. When switching to a public cloud you are changing the way the interact with your customers and how your employees operate on their day to day.

Include comparisons of how the cloud is better than on-premise stuff

\*conclusion -hybrid cloud structure\*

<https://aws.amazon.com/solutions/case-studies/bankinter/> (no.1)

<https://www.spglobal.com/marketintelligence/en/news-insights/latest-news-headlines/as-big-tech-dominates-cloud-use-for-banks-regulators-may-need-to-get-tougher-59669007>

<https://aip.scitation.org/doi/pdf/10.1063/1.5014007>

<https://aws.amazon.com/blogs/industries/dealing-with-disruption-how-financial-institutions-are-using-cloud-technology-to-respond-to-covid-19-and-reshape-the-industry/>

<https://aws.amazon.com/solutions/case-studies/conflux/> (no.2)

<https://www.spiceworks.com/finance/fintech/guest-article/banking-in-public-cloud-heres-what-financial-institutions-need-to-consider/>

<https://www.bain.com/insights/countering-the-myths-that-hinder-cloud-adoption-in-financial-services/>

<https://www.10thmagnitude.com/opex-vs-capex-the-real-cloud-computing-cost-advantage/>

<https://qualitestgroup.com/insights/blog/cloud-migration-3-biggest-risks-banks-and-financial-services-companies-need-to-know-and-how-to-avoid-them/>

**Task 2**

**Task A:**

S3 Bucket names:

tjc-wordfreq-dec3-uploading ------ uploading bucket ---- arn:aws:s3:::tjc-wordfreq-dec3-uploading

tjc-wordfreq-dec3-processing ------ processing bucket ---- arn:aws:s3:::tjc-wordfreq-dec3-processing

s3://tjc-wordfreq-dec3-uploading/lsde-wordfreq-app/lsde-wordfreq-app.zip --- zip file S3 URL

SQS Names:

wordfreq-jobs ---- <https://sqs.us-east-1.amazonaws.com/657190762770/wordfreq-jobs>

wordfreq-results ---- <https://sqs.us-east-1.amazonaws.com/657190762770/wordfreq-results>

SNS Names:

wordfreq-file-copied ---- arn:aws:sns:us-east-1:657190762770:wordfreq-file-copied

aws s3 cp s3://tjc-wordfreq-dec3-uploading s3://tjc-wordfreq-dec3-processing --exclude "\*" --include "\*.txt" --recursive

aws s3 cp s3://zj-wordfreq-nov22-uploading s3://zj-wordfreq-nov22-processing --exclude "\*" --include "\*.txt" --recursive

**Task B**

**How the application works**

The basic functionality of the Wordfreq application is that the top 10 most frequent words are displayed by reading through a text file and counting the number of times each word appears. The way the wordfreq app works in AWS is that firstly two s3 buckets are created that store all of the the text files that will be used. One is an uploading bucket which will store the files on my local machine and the other is a processing bucket that will process these files. When the application is run, the files will be copied from the uploading bucket to the processing bucket. There are also two SQS queues used in this application, one is a jobs queue which holds the notification messages of the uploaded text files in the s3 bucket, and the other queue is a results queue that holds the messages containing the top 10 most frequent words in each specific text file. SNS is also used in this application but is not required for the application to work but is helpful for receiving emails for when files are uploaded and processed so you can see the application is working correctly. A DynamoDB database is used for storing the top 10 results in a NoSQL database which can be easily viewed and each file is displayed with key:value pairs showing the name of the word and the number of times it appears in that file. Finally, the application is run on a specific EC2 Ubuntu instance that is the compute power for this application and currently uses a t2.micro instance type but can be changed to increase compute power for faster run times of the application. In order to run the application you must first connect to the EC2 instance using a SSH connection and then run the command: “aws s3 cp s3://tjc-wordfreq-dec3-uploading s3://tjc-wordfreq-dec3-processing --exclude "\*" --include "\*.txt" –recursive” where it will take approximately 10 minutes at default to complete.