Coursework - EMATM0051 Large Scale Data Engineering [Data Science cohort]

[**Note**: If you are NOT from MSc Data Science Program, this coursework is not for you. Please contact your unit director immediately.]

Summary

This coursework is divided into two parts:

Part 1: A written task (only) to design the architecture of a simple application on AWS cloud, where you are required to have a deep understanding of AWS services and how they work together within an application. The design should demonstrate your knowledge of AWS services covered throughout the entire LSDE course.

Part 2: A combined practical and written activity architecting a scaling application on the Cloud, where you will be required to use the knowledge gained and a little further research to implement the scaling infrastructure, followed by a report that will focus on your experience in the practical activity together with knowledge gained in the entire LSDE course.

You should use AWS Academy Learner Lab [142821] for this coursework (Part 2).

Weighting: This assessment is worth 100% of your total unit 20 credits.

Due: 13:00. Tue 9th Dec 2025.

Please note that the Category of generative AI use in assessment for this assessment falls in **Category 2: Minimal** – for example, using spell and grammar checkers to help identify mistakes but not rewrite chunks of text. More information is available on https://www.bristol.ac.uk/students/support/academic-advice/using-artificial-intelligence/#categories

Please note that all information shown in the screenshots must be in **English**. The screenshots will be considered invalid if they include any text in non-English languages. **PDF Format Warning**: The submitted report must be in a text-based PDF format. If your pdf file is image-based, it will prevent Turnitin from performing similarity and AI checks. A report in an image/figure format will not be accepted.

Pre-requisites:

- You must have completed the AWS Academy Cloud Foundations course set in weeks 1-7
- You will require an AWS Academy Learner Lab account for the practical activity. You should receive an invite when this document is released. Please contact the LSDE Unit Director if you don't receive email or you have issues with the registration.
- A Secure Shell (SSH) client, such as MacOS Terminal or PuTTy on Windows, for server admin.

Submission:

Via the LSDE Blackboard coursework assessment page, submit one pdf file, named using your UOB username ('username.pdf), containing:

- A report in PDF format containing:
    - Part 1
    - Part 2
    - Your AWS Academy account credentials (username, password)

In this document we provide a detailed explanation of the tasks, and the approach to marking.

[Note: If you cannot find the submission point, please email SEMT Student Enquiries Mailbox (semt-student-enquiries@bristol.ac.uk) immediately.]

Unit Director: Jin Zheng

Part 1: (25%)

In Part 1, we require you to design the AWS architecture for a public webservice called "ArtAI", where users can upload images and receive an AI-generated variation (for example, a stylised or enhanced version of their photo). Your architecture should ensure the service is reliable, secure, and scalable.

The application needs to meet the following requirements:

- Provide low-latency and highly available access to the site worldwide.
- Support user uploads of images and process them using a pre-trained AI model for inference. Assume the model is already well-trained, so no training is required. You will need to choose an AWS service to deploy the model and justify your choice. The AI model must only be accessible by the application backend. It should not be directly exposed to the public, as you do not want to release the model.
- Ensure that all uploaded and generated images are backed up automatically, and earlier versions can be restored if needed.
- Trigger an email notification if the number of requests exceeds 10,000 in a single day.

You should include your own descriptions of the following, 500-800 words and no more than 2 A4 pages:

- List the AWS services used in your design and explain in detail how these services work to ensure the **high-performance, security and cost-efficiency** in this application.
- Use a diagram to demonstrate the architecture of this application, especially for showing AWS services interaction and your network design. Please also describe how this application works to back up your data automatically when you upload them.

[**Note**: Ensure your diagram is clear, well-labeled, and professionally presented. You are strongly recommended to use diagrams.net (also known as draw.io, https://app.diagrams.net/) or a similar professional tool to create your diagram. Hand-drawn drafts or AI-generated figures will not be accepted.]

You don't need to implement these ideas in your lab account.

Part 2: Scaling the WordFreq Application (75%)

Write a report of no more than 2500 words and 18 A4 pages (there is NO minimum), including: Task A, B, C, D, E.

Overview

WordFreq is a complete, working application, built using the Go programming language.

*[**NOTE:** you are **NOT** expected to understand or permitted to modify the source code in any way]*

The basic functionality of the application is to count words in a text file. It returns the top ten most frequent words found in a text document and can process multiple text files sequentially.

The application uses a number of AWS services:

- S3: There are two S3 buckets used for the application.
    - One is used for uploading and storing original text files from your local machine. This is your uploading bucket.

    - These files will be copied from the uploading bucket to the processing S3 bucket. The bucket has upload notifications enabled, such that when a file is uploaded, a message notification is automatically added to a wordfreq SQS queue.
- SQS: There are two queues used for the application.
    - One is used for holding notification messages of newly uploaded text files from the S3 bucket. These messages are known as 'jobs', or tasks to be performed by the application, and specify the location of the text file on the S3 bucket.
    - A second queue is used to hold messages containing the 'top 10' results of the processed jobs.
- DynamoDB: A NoSQL database table is created to store the results of the processed jobs.
- EC2: The application runs on an Ubuntu Linux EC2 instance, which you will need to set up initially following the instructions given. This will include setting up and identifying the S3, SQS and DynamoDB resources to the application.

You will be required to initially set up and test the application, using instructions given with the zip download file. You will then need to implement auto-scaling for the application and improve its architecture based on principles learned in the CF course. Finally, you will write a report covering this process, along with some extra material.

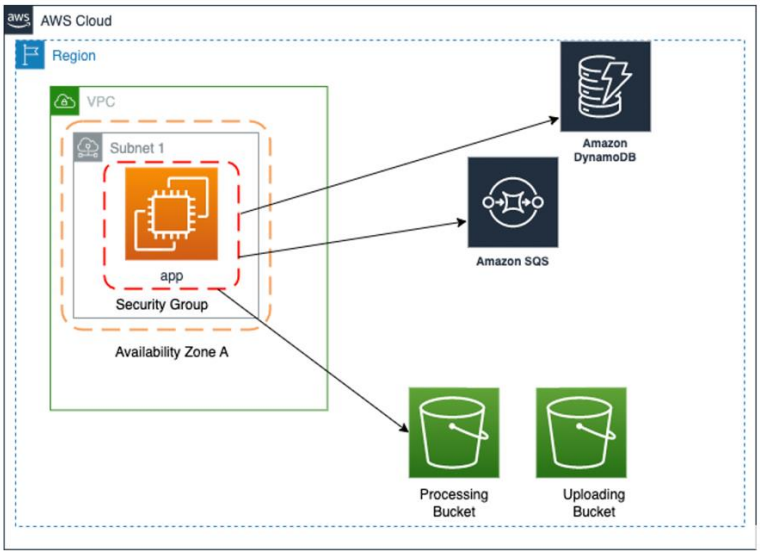

Figure 1 - WordFreq standard architecture

Task A – Install the Application

Ensure you have accepted access to your AWS Academy Learner Lab account and have at least $10 credit (you are provided with $50 to start with). If you are running short of credit, please inform your instructor.

Refer to the WordFreq installation instructions ('README.txt') in the coursework zip download on the Blackboard site, to install and configure the application in your Learner Lab account. These instructions do not cover every step – you are assumed to be confident in certain tasks, such as in the use of IAM permissions, launching and connecting via SSH to an EC2 instance, etc.

You will set up the database, storage buckets, queues and worker EC2 instance. Finally, ensure that you can upload a file and can see the results logged from the running worker service, before moving on to the next task.

You will need to give a brief summary of how the application works (without any reference to the code functionality) in this Task.

*[**NOTE:** The application code is in the Go language. You are **NOT** expected to understand or modify it. Any code changes will be ignored and may lose marks.]*

Task B – Design and Implement Auto-scaling

Review the architecture of the existing application. Each job process takes a random time to complete between 10-20 seconds (artificially induced, but DO NOT modify the application source code!). To be able to process multiple uploaded files, we need to add scaling to the application.

This should initially function as follows:

- When a given maximum performance metric threshold is exceeded, an identical worker instance is launched and begins to also process messages on the queues.
- When a given minimum performance metric threshold is exceeded, the most recently launched worker instance is removed (terminated).
- There must always be at least one worker instance available to process messages when the application architecture is 'live'.
- You want to ensure that there is a two-minute interval between each new instance launch.

Using the knowledge gained from the Cloud Foundations course, architect, **please implement auto-scaling functionality for the WordFreq application and demonstrate how you configure the auto-scaling policy.** Note that this will not be exactly the same as Lab 6 in Module 10, which is for a web application. You will not need a load balancer, and you will need to identify a different CloudWatch performance metric to use for the 'scale out' and 'scale in' rules. The 'Average CPU Utilization' metric used in Lab 6 is not necessarily the best choice for this application.

Task C - Perform Load Testing

Once you have set up your auto-scaling infrastructure, test that it works. The simplest method is to create around 130 text files. You could use the text files on Blackboard. Please make sure you've uploaded all 130 files to your uploading S3 bucket before starting this task.

You can 'purge' all files from your processing S3 bucket, then you could copy all the .txt files from you uploading S3 bucket to your processing S3 bucket. Please stop the original instance wordfreq-dev and only use the instances that are created by your auto scaling group.

- Connect to one of your instances that in your Auto Scaling Group (via SSH connection).
- Copy all the .txt file from your uploading S3 bucket (e.g., zj-wordfreq-nov25-uploading) to your processing S3 bucket (e.g., zj-wordfreq-nov25-processing) by running the following command in your SSH terminal:
  aws s3 cp s3://<name of your uploading bucket> s3://<name of your processing bucket> --exclude "*" --include "*.txt" --recursive

Please watch and record the following behaviours and illustrate all loading tests done for optimising auto-scaling:

- Watch the behaviour of your application to check the scale out (add instances) and scale in (remove instances) functionality works.

- Take screenshots of your copied files in the S3 bucket, the SQS queue page showing message status, the Auto Scaling Group page showing instance status, the EC2 instance page showing launched / terminated instances and the output from DynamoDB during this process.
- Try to optimise the scaling operation, for example so that instances are launched quickly when required and terminated soon (but not immediately) when not required. Note down settings you used and the fastest file processing time you achieved.
- Try using a few different EC2 instance types – with more CPU power, memory, etc. Please record the processing time for each experiment and discuss your findings.

***[NOTE:** Please delete all the .txt file in your processing S3 bucket after load testing]*

***[NOTE:** Ensure that your WordFreq application's auto-scaling is still functional when finished!]*

[NOTE: The Learner Lab accounts officially only allow a maximum of 9 instances running in one region, including auto-scaling instances**. Learner Lab accounts are* Limited *in which EC2 Types and AWS services they can use. This is explained in the Lab Readme file on the Lab page; section 'Service usage and other restrictions'**. Please note that you may get your account deactivated if you attempt to violate the service Restrictions

Task D - Optimise the WordFreq Architecture

Based on **only** AWS services and features learned from the Cloud Foundations course, describe how you could re-design the WordFreq application's current cloud architecture (i.e. not changing the application's functionality or code) to improve the architecture in the following areas:

- Increase resilience and availability of the application against component failure.
- Long-term backups of valuable data required.
- Cost-effective and efficient application for occasional use. Processing does not need to be immediate.
- Prevent unauthorised access.

Your description should ideally include diagrams and include the AWS services required together with a high-level explanation of features & configuration for each requirement.

[**Note**: Ensure your diagram is clear, well-labeled, and professionally presented. You are strongly recommended to use diagrams.net (also known as draw.io, https://app.diagrams.net/) or a similar professional tool to create your diagram. Hand-drawn drafts or AI-generated figures will not be accepted.]

You don't need to implement these ideas in your lab account.

Task E – Further Improvements

Based on services and frameworks covered in the full LSDE course, identify two alternative data processing services that would be far more performant and robust for this processing task. Please describe their advantages over the current version of WordFreq in a few paragraphs.

You don't need to implement these ideas in your lab account.

Final Task

Combine Part 1 and Part 2 to a single PDF. **You will also need to give us Your AWS Academy account credentials (username, password) at the end of your report.**

The report should be submitted as a single PDF and adhere to the following format:

- Page limits: Part 1 has a maximum of 2 pages and Part 2 a maximum of 18 pages. These limits are strict and include all tables, figures, and references.
- Font Size: Minimum 11 pt for all text, including footnotes and captions.
- Margins: 2.54 cm (1 inch) on all sides (top, bottom, left, right).
- Line Spacing: Single spacing.
- Structure: Write in a clear and organized manner, using paragraphs and sub-headings effectively.
- Tables and Figures: All tables and figures must be properly labelled with titles and, if necessary, brief descriptions. Ensure that all figures are clear and legible, with information easily readable without zooming. The titles of tables/figures and their descriptions in the caption will not be included into words count.
- Appendices: Do not include appendices, as they will not be reviewed. All content, including figures and tables, must appear in the main text.

[IMPORTANT: All text not originally created by you must be cited, leading to a final numbered reference section (based on e.g. the British Standard Numeric System) to avoid accusations of plagiarism.]

[IMPORTANT: Disable autoscaling at end of each lab session: – Desired capacity = 0 ; Minimum capacity = 0. This saves credit and avoids multiple instances from launching and terminating when starting / stopping a lab session]

AWS Academy Learner Lab

You are given an AWS Academy Learner Lab account for this coursework. Each account has $50 assigned to it, which is updated every 24 hours and displayed on the Academy Lab page.

To access the lab from AWS Academy, select *Courses > AWS Academy Learner Lab [142821]> Modules > AWS Academy Learner Lab> Launch AWS Academy Learner Lab.* On this page click 'Start Lab' to start a new lab session, then the 'AWS' link to open the AWS Console once the button beside the link is green.

Please note:

- Ensure you shut down (stop or terminate) EC2 instances when you are not using them. These will use the most credit in your account in this exercise. Note that the Learner Lab will stop running instances when a session ends, then restart them when a new session begins.
- AWS Learner Lab accounts have only a limited subset of AWS services / features available to them, see the Readme file on the Lab page (Service usage and other restrictions).
- If you have installed the AWS CLI on your PC and wish to access your Learner Lab account, you will need the credentials (access key ID & secret access key) shown by pressing the AWS Details button on the Lab page. Note that these only remain valid for the current session.
- If you have any issues with AWS Academy or the Learner Lab, please book an Office Hours session or use the LSDE Discussion Forums to seek help FIRST, email the instructors if there is no other option.

Marking

Below are the marking bands with maximum possible mark range achievable given approximate scope of work.

+80%: Outstanding report and implementation. Extensive exploration, analysis and implementation demonstrating deep understanding and reading outside of the CF course and lectures.

70 - 80%: Excellent report. Well architected, fully functional auto-scaling, great optimisation techniques, very good understanding of cloud principles gained in the CF course.

60 - 70%: Report of correct length, fully functional auto-scaling, good optimisation techniques, good understanding of cloud principles gained in the CF course.

50 - 60%: Report of correct length, basic but functional auto-scaling, some good ideas about optimisation techniques, correct understanding of main cloud principles in the CF course.

<50% (Fail): Report is not at an appropriate standard, auto-scaling not implemented. Objectives of the assignment have not been demonstrated.

Academic Offences

Academic offences (including submission of work that is not your own, falsification of data/evidence or the use of materials without appropriate referencing) are all taken very seriously by the University. Suspected offences will be dealt with in accordance with the University's policies and procedures. If an academic offence is suspected in your work, you will be asked to attend an interview with senior members of the school, where you will be given the opportunity to defend your work. The plagiarism panel are able to apply a range of penalties, depending the severity of the offence. These include: requirement to resubmit work, capping of grades and the award of no mark for an element of assessment.

Extensions and Exceptional Circumstances

If the completion of your assignment has been significantly disrupted by serious health conditions (including mental health impairment), personal problems, or other similar issues, you may be able to apply for an extension for assessment submission or consideration of extenuating circumstances (in accordance with the normal university policy and processes). **Please check with your personal tutor as the LSDE teaching team won't be able to help with it.**

- Extensions allow limited additional time to be granted before submission. They must be requested before the normal assessment submission date. See the following page: https://www.bristol.ac.uk/students/support/academic-advice/assessment-support/request-a-coursework-extension/. Note that all assessment extension requests in AY 2023-24 require evidence.

- Exceptional Circumstances (EC) recognises a significant disruption and can facilitate extensions, additional support and care services, waiving of late submission penalties, extension of studies, etc. Students should contact the LSDE Unit Director and their tutor and apply for consideration of EC as soon as possible when the problem occurs. Please review the following university page: https://www.bristol.ac.uk/students/support/academic-advice/assessment-support/extenuating-circumstances/