# Problem Identification and Justification

## Problem 1: Automate Cloud Resource Usage Tracking

## Description:

## Management and optimization of resource usage are the most important things to do in cloud computing. One of the problems involves the writing of a Python script that will be used in cloud computing in monitoring cloud resource usage like CPU and memory utilization, sending out alerts whenever any threshold value has been exceeded. Monitoring such usage proactively supports a business in its resource allocation and may prevent performance degradation or budget overruns.

## Application:

## This script is of great help to cloud administrators and developers responsible for maintaining applications on platforms such as AWS, Azure, or Google Cloud Platform. They can monitor resource usage and manage it more effectively, which will be really beneficial to any organization working on a tight budget or whose application workloads tend to become quite unpredictable. This may include alerts when usage crosses the threshold set, which could trigger early adjustment in efforts toward diminishing wasteful cloud spending while ensuring that performance remains at its best.

## Difficulty:

## This project is moderately challenging. Most analysis on cloud resource usage usually has to be done through API requests, such as AWS' boto3 SDK or Azure's azure-mgmt library. Setting up API access and parsing API response data requires a little more foundational information about cloud computing and authentication. However, the meat of the problem-tracking values and comparing them to thresholds-can fit into an introductory Python framework.

## Python Libraries:

## The main ones needed are cloud SDKs, e.g.,

## boto3: For AWS, it fetches cloud metrics like CPU and memory utilization.

## requests: A general REST APIs library

## smtplib: Optional; useful for sending email alerts directly from the script.

## These, in turn, will enable the script to periodically fetch data from the cloud environment, evaluate it, and send notifications in the event of any threshold crossing. The project applies control structures-for example, loops and conditionals-and function definitions to obtain modularity.

## Problem 2: Organize Cloud Log Files for Easier Access and Review

Description:

Write a Python script that will be able to auto-organize cloud log files in their layout against certain parameters like date or log level. Such logs provided by the cloud service are very useful for debugging error tracking and other performance monitoring issues. If big volumes exist, using them to dig out data as soon as possible is quite a problem. This would scan a directory containing cloud logs, sort them in order of date or any other given criteria, and save nicely organized copies to a new directory.

Application:

It will increase the speed of debugging; otherwise, it would have taken ages and system monitoring. Scattered log files everywhere would no longer pose any big problem since the users would access easy-to-navigate and well-organized files rather than hunting. This is quite an important tool to facilitate access to logs, save time in log management; hence, it is quite handy for cloud administrators, developers, and IT teams.

Problem

This is slightly easier compared to the cloud resource usage tracker. This, at its core, is a problem of file management where one needs to rename, move, and sort files. This is good for someone who knows the basics of Python because most of the purpose can be served by built-in libraries.

Libraries Used:

The most important library that will be used for this purpose is

os: It can allow accessing to the operating system to rename/move a file.

datetime: Allows for date extraction and formatting so files sort by date.

logging (optional): Allows for a user-created log file so the actions that occur in the script are summarized such as "File moved" or "Error reading file.".

Functions are in place for the division of what the script does-for logical control structure flows and standard libraries in Python to maintain files and directories. This can also be extended in code so that it can filter out those files that have been created before a certain date.