Product Knowledge

Planning Exercise

Introduction

In this exercise, you will demonstrate your ability to describe the functionality and purpose of the monitoring, logging, and observability products within Google Cloud. In Part 1, you will summarize the functionality of each of the listed products. In Part 2, you will match these products to business and technical requirements.

Part 1

In the space provided, briefly summarize the business and/or technical benefits of each of the listed products.

Product	Business and/or Technical Benefits
Monitoring	
Logging	
Incident Response and Management	
Error Reporting	
Service Monitoring	

Debugger

Trace

Profiler

Part 2

Each of the scenarios below describe a business or technical need which may be satisfied by one of the above listed monitoring, logging, and observability products in Google Cloud.

Although some of the needs below may be satisfied by more than one product, for the purposes of this exercise please select the one (1) product that is best suited to satisfy the requirements.

Note that products may be used multiple times.

Business or technical need	**Best suited product**
Calculate the uptime of a service and receive alerts if the uptime value falls below a certain threshold.	
Your CIO would like a visual aid placed in the network operations center to showcase the amount of traffic to your web applications in real-time.	
Perform a periodic review of the administrative actions taken within GCP.	
An application is generating errors, and you suspect firewall rules may be the culprit.	
Your network security team wishes to analyze the network traffic within GCP for suspicious behavior.	
Your customers are posting on social media about the unreliability of your web services, yet your helpdesk has reported very few trouble tickets from customers.	
You notice that the number of people using your web application has dropped significantly after the last few updates to the application, but no one is complaining.	
Users have reported that an application occasionally returns garbage data instead of the intended results, but you have been unable to reproduce this problem in your test environment.	
Your geographically distributed development team is working on a complex and difficult to explain bug and could benefit from more collaboration during their work.	

The execution of your production application is sluggish, yet you fully measured the code-performance in your test environment.	
Your developers are looking for ways to improve the performance of their code, and need greater insight into the resource utilization incurred by different programming approaches within their code.	
Due to unexpected demand, you have increased the scale of your application deployment, however the application performance is not improving. You suspect a bottleneck somewhere.	
You closely monitor the latency of your applications for sudden changes that degrade performance. Your applications are regularly updated, and you want to be informed if these updates cause subtle latency changes over time.	
You are trying to find the source of seemingly random time out errors within a large microservices application.	
You need a tool to help identify the dependencies within a distributed microservices application with hundreds of interconnected services.	