# Google Cloud Functions Use Case

## Use Case: Automating Event-Driven Workflows with Google Cloud Functions

### GENERAL CHARACTERISTICS

| | |
|---|---|
| Intent | To automate event-driven workflows for a client using Google Cloud Functions. |
| Scope | Implementation of serverless functions to trigger automated workflows based on specific events. |
| Level | System-level. |
| Client | Confidential (E-Commerce Platform). |
| Last Update | 03/12/2024 |
| Status | Finalized. |
| Stage | Implementation and Optimization. |

### ACTORS

| | |
|---|---|
| Primary Actor | Cloud Engineer. |
| Secondary Actors | DevOps Team, Application Developers, Operations Team. |

### PREREQUISITES

| | |
|---|---|
| Static Preconditions | - Google Cloud Project set up with Cloud Functions API enabled.<br>- Events and triggers defined for workflow automation. |
| Dynamic Preconditions | - Dependencies packaged for serverless deployment.<br>- IAM roles and permissions configured for secure function execution. |
| Assumptions | - Client requires scalable and cost-effective solutions for automating workflows.<br>- Workflows must be triggered in response to events like file uploads, API calls, or database updates. |

### TRIGGERS

| | |
|---|---|
| Trigger Event | The client needed an automated solution to process real-time file uploads and update their inventory database. |

# Google Cloud Functions Use Case

## EXPECTED OUTCOME

| | |
|---|---|
| Success Postcondition | - Workflows are triggered and executed automatically in response to events.<br>- Operational efficiency is improved with reduced manual intervention. |
| Failed Postcondition | - Events fail to trigger workflows, causing delays in processing tasks. |

## OPERATIONS AND CONCEPTS

| | |
|---|---|
| Operations | 1. Created a Cloud Function to process file uploads from Cloud Storage.<br>2. Configured an event trigger to execute the function whenever a file is uploaded.<br>3. Integrated the function with Firestore to update inventory data in real-time.<br>4. Tested the function for scalability and ensured it could handle peak traffic.<br>5. Set up monitoring and logging for troubleshooting and performance insights. |
| Concepts | - Cloud Functions: Serverless execution environment for building and connecting services.<br>- Event Triggers: Automatically execute functions in response to specific events.<br>- Integration: Seamless interaction with other GCP services like Cloud Storage and Firestore. |

## MAIN SUCCESS SCENARIO

| | |
|---|---|
| Step 1 | Identified key events requiring automation, such as file uploads. |
| Step 2 | Developed a Cloud Function to process uploaded files and extract necessary data. |
| Step 3 | Configured Cloud Storage as the trigger source for the function. |
| Step 4 | Integrated the function with Firestore for real-time inventory updates. |
| Step 5 | Enabled monitoring to ensure the function executed successfully. |
| Step 6 | Scaled the function automatically to handle peak traffic during promotional events. |
| Step 7 | Reduced manual intervention by automating workflows and minimizing errors. |