

Leveraging StackDriver for Monitoring Cloud Functions



Janani Ravi

CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

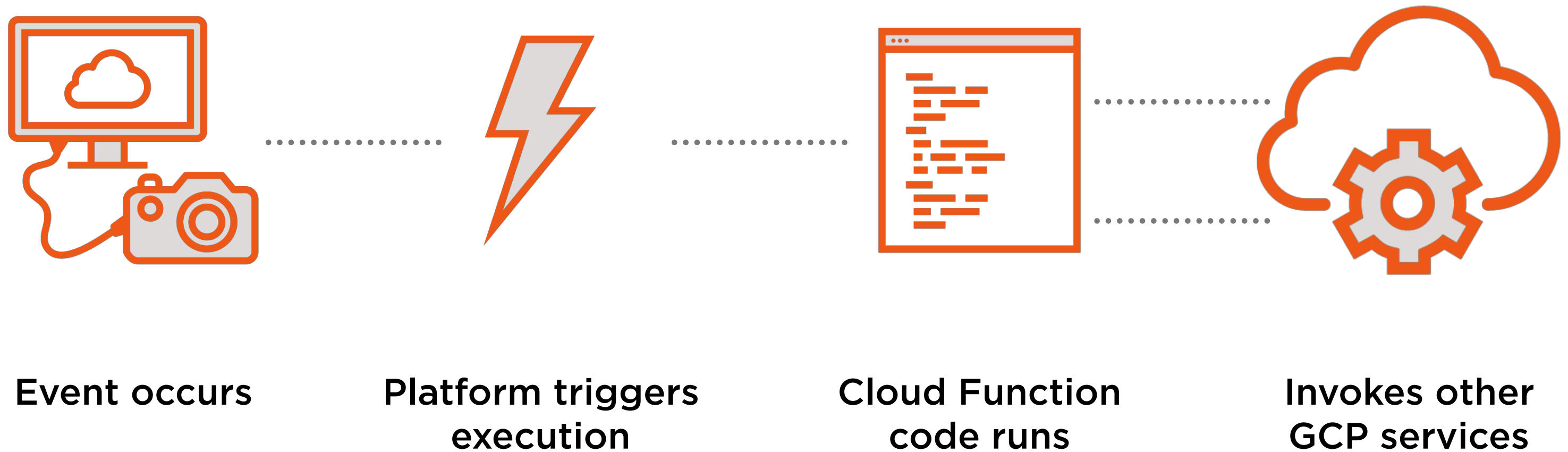
StackDriver for monitoring Cloud Functions

Working with logs from within Cloud Functions

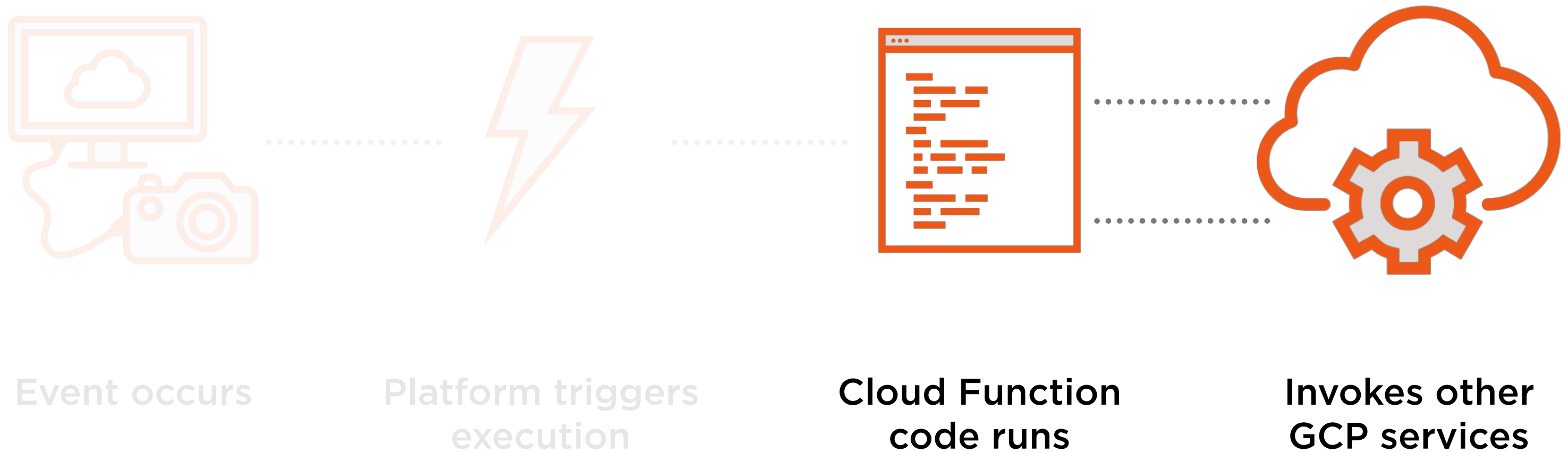
Reporting errors from within Cloud Functions

Retrying Cloud Functions

Event-driven Serverless Compute



Event-driven Serverless Compute





Use Stackdriver suite with Cloud Functions

- Capture/store logs
- Capture/display formatted error messages
- Record execution metrics

Demo

View Cloud Function logs

Reading logs using Cloud Functions

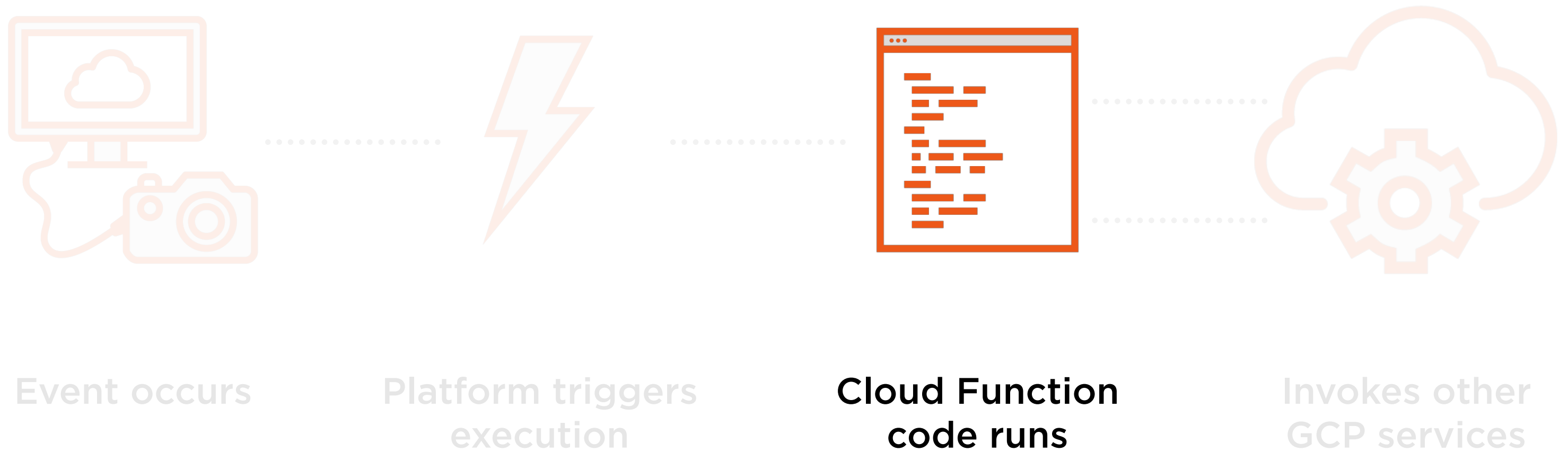
Demo

Using Stackdriver monitoring graphs

Configuring alerts in Stackdriver

Best Practices

Event-driven Serverless Compute



Execution Environments

Currently, limited runtimes

Python

- Python 3.7.0
- Flask to handle requests

JavaScript

- Node.js 6.14.0 default
- Node.js 8.11.1 beta





Idempotence

Function should work even if invoked multiple times

Allows re-invocation

Graceful handling of failed invocation



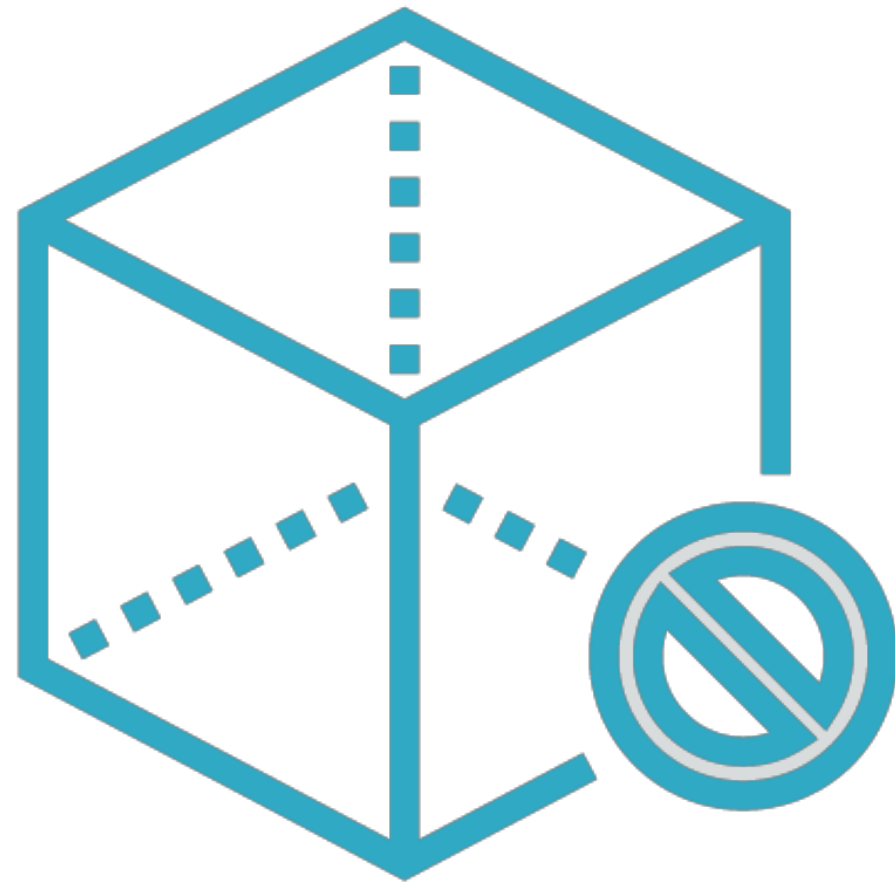
Avoid Timeouts

How? Signal that function code is complete

Why? You are charged for entire period of timeout

Timeouts may also cause re-invocation (from cold start)

Avoid Background Activities



No activity should occur after function has terminated

Can lead to unexpected, hard-to-spot errors

Identify background activities from logs

Delete Temporary Files

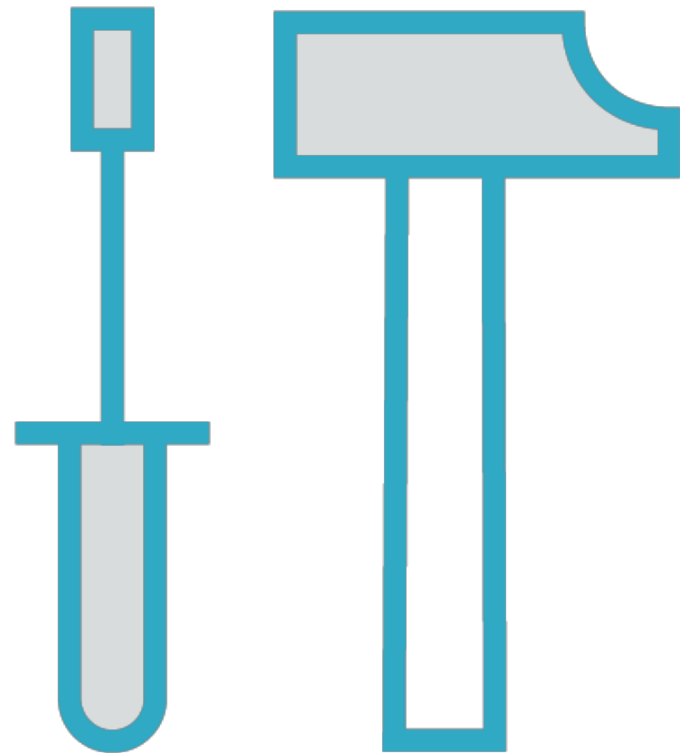


Local disk storage is in-memory

Failure to delete can cause out-of-memory errors

Might sometimes persist between invocations

Other Best Practices

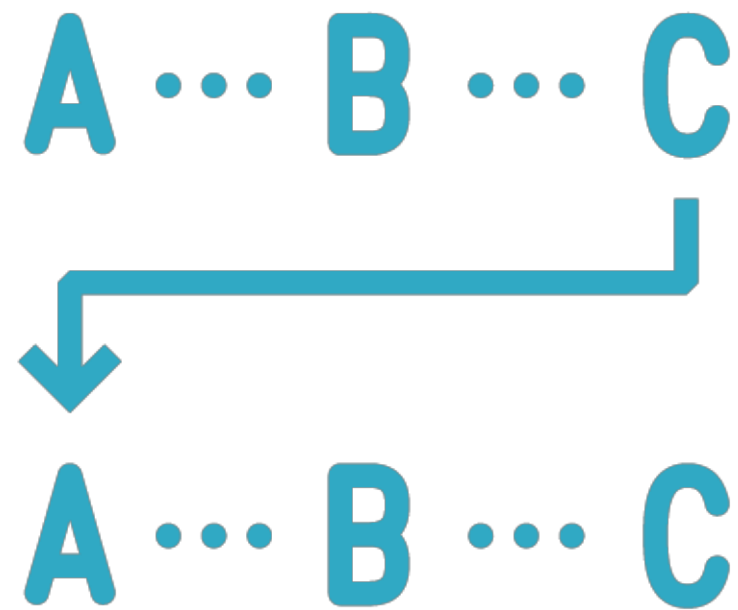


Test function locally to save time, money

Do not throw uncaught exceptions

Use dependencies wisely for faster startup

Retrying Cloud Functions

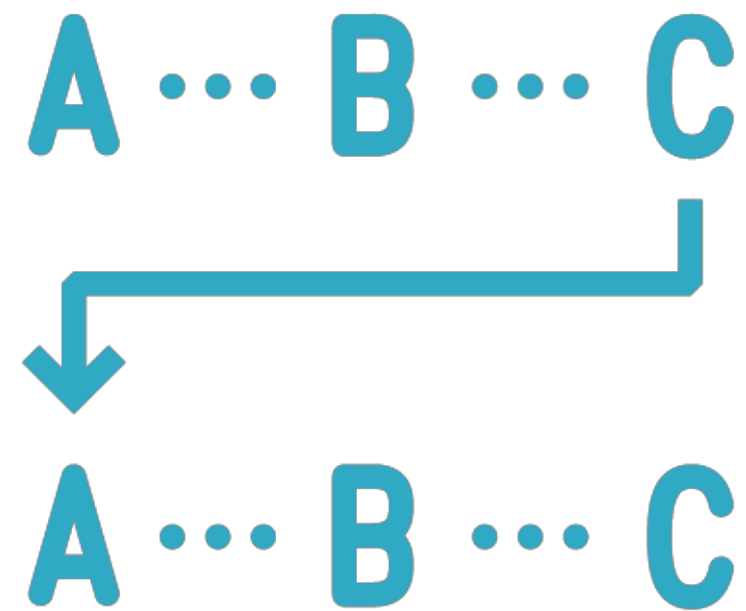


Best-effort semantics

- Default configuration
- Execution is not guaranteed

At-least-once semantics

- Need to enable retries
- Repeated failures may still time out

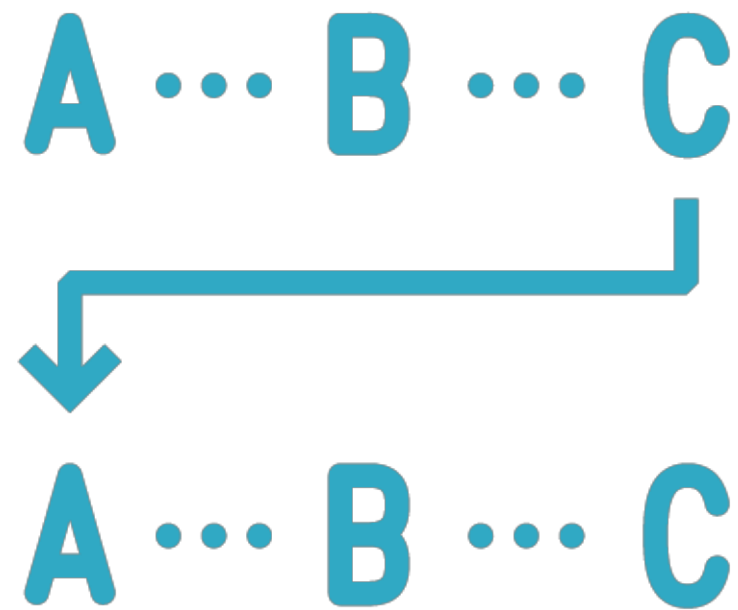


“Retry on failure”

Use only with tested, hardened functions

- Follow best practices

Else stuck in retry loop



Only to handle transient errors

Set end condition to avoid infinite retry loops

Distinguish between fatal and non-fatal errors

Make function idempotent

Demo

**Configuring retries on background
Cloud Functions**

Demo

Error reporting

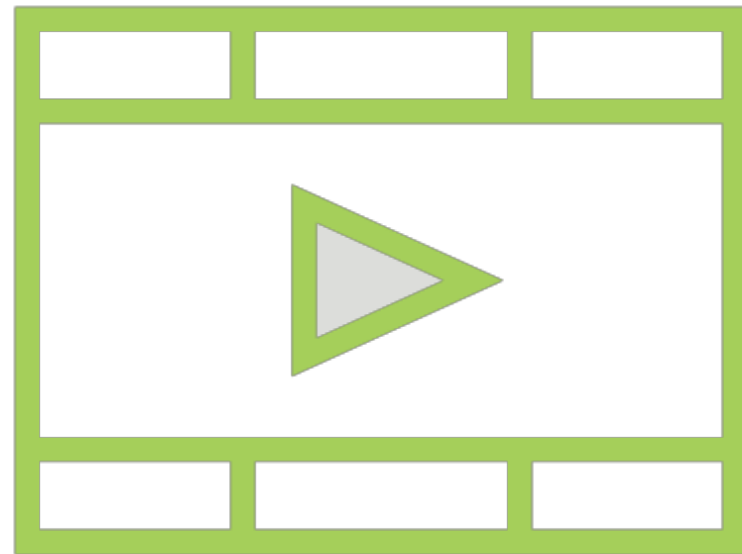
Summary

StackDriver for monitoring Cloud Functions

Working with logs from within Cloud Functions

Reporting errors from within Cloud Functions

Retrying Cloud Functions



Related Courses

Building Chatbots with Google Dialogflow

AWS Developer - An Introduction to AWS Lambda

Microsoft Azure Developer: Create Serverless Functions