**Dokumentation Killswitch**

Inhaltsverzeichnis

[Planung: 1](#_Toc143624860)

[Auslesen der Log-Nachricht: 2](#_Toc143624861)

[Erstellung des Budgets: 2](#_Toc143624862)

[Budgetabfrage: 3](#_Toc143624863)

[Billing Abfrage: 4](#_Toc143624864)

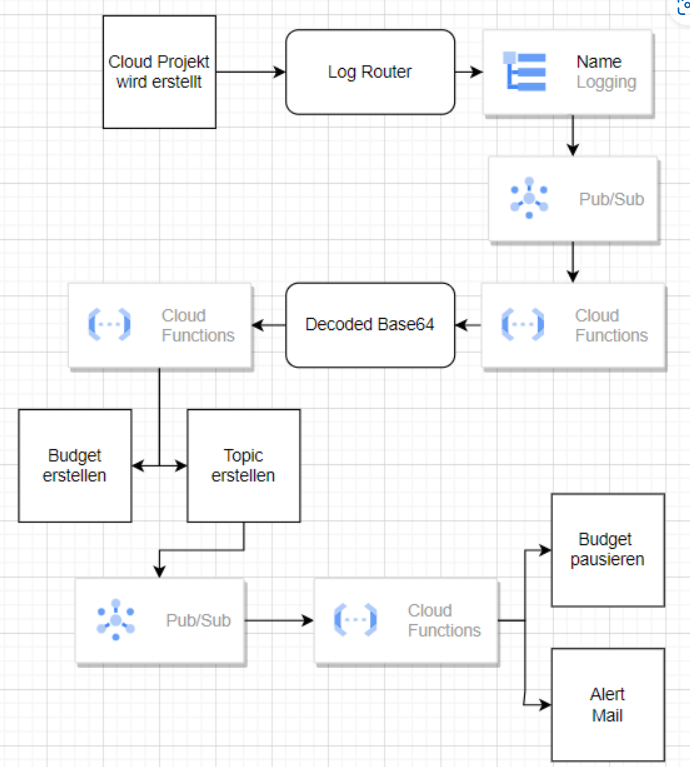
[Disable Billing: 4](#_Toc143624865)

[Alert Mail: 5](#_Toc143624866)

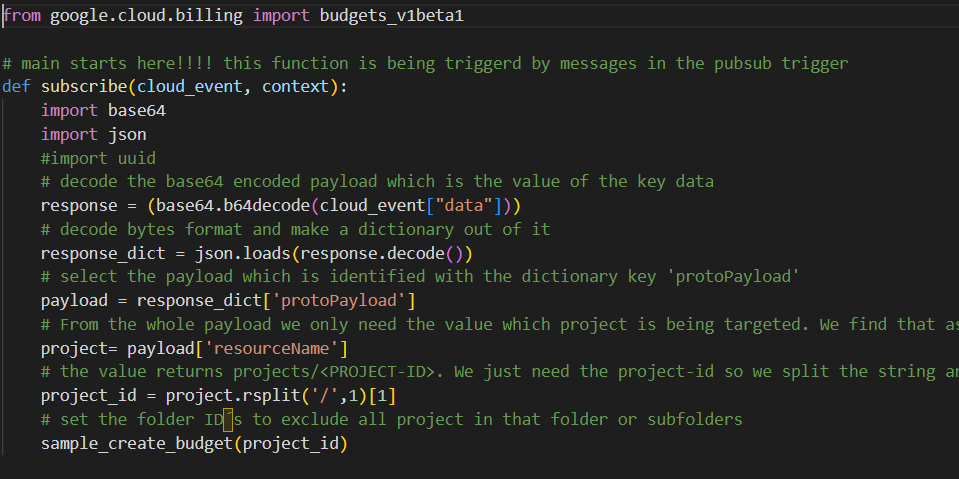
# Planung:

Das Ziel des Projektes ist es ein Killswitch für eine Cloudumgebung zu entwickeln der alle kostenerzeugenden Objekte bei Überschreiten eines Budgets deaktiviert.

Der Ablauf beginnt mit der Erstellung eines Projektes, welches einen Trigger via Pub/Sub auslöst und somit eine Cloud Function aufruft. Die Cloud Function liest daraufhin die entstandene Log-Nachricht aus. Diese Daten aus der entschlüsselten Base64 Datei werden dann an eine weitere Function weitergegeben, welche damit ein Budget erstellt. Dieses Budget ist mit einem Topic verbunden, welches dann einen Pub/Sub Trigger auslöst. Sobald dieser Trigger durch Budgetüberschreitung aktiviert wird, wird durch die letzte Cloud Function eine Alert-Nachricht an alle Kontakte, die angegeben sind, geschickt und alle Kosten erzeugenden Objekte werden heruntergefahren.



# Auslesen der Log-Nachricht:



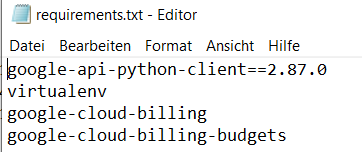
Nach dem Erstellen eines Projektes wird diese Funktion aufgerufen. Die Funktion entschlüsselt die Base64 Datei und gibt die Projekt-ID aus.

# Erstellung des Budgets:

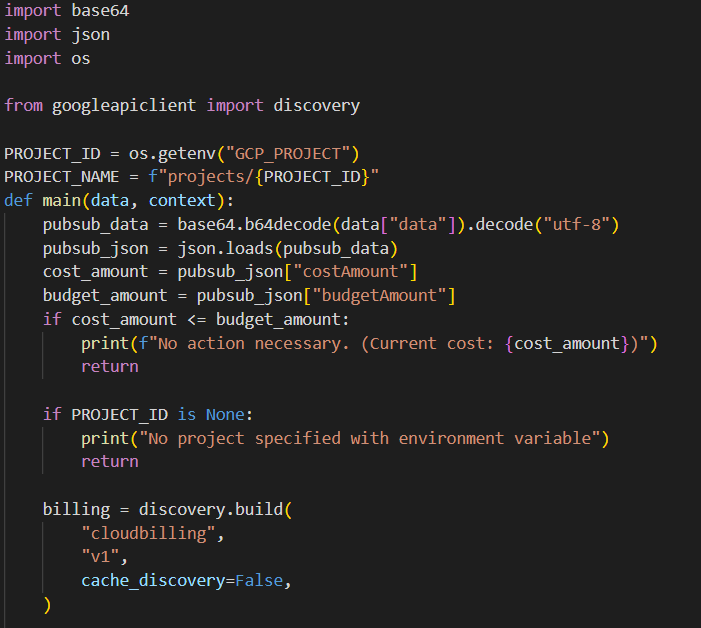


Für diese Methode muss man die Budget-Vorlage „v1beta1“ von Google Cloud Billing importieren. Aus dieser Vorlage wird zuerst der Service Client aufgerufen und danach werden in der „request“ die Einstellungen für das Budget definiert. Dort kann man z.B. die Währung oder die Höhe für das Budget festlegen. Daraufhin wird auf Basis dieser Anforderungen ein Budget erstellt.

Man benötigt zusätzlich für die Ausführung der Methode eine Text-Datei mit dem Namen „requirements“ mit den folgenden Inhalten:

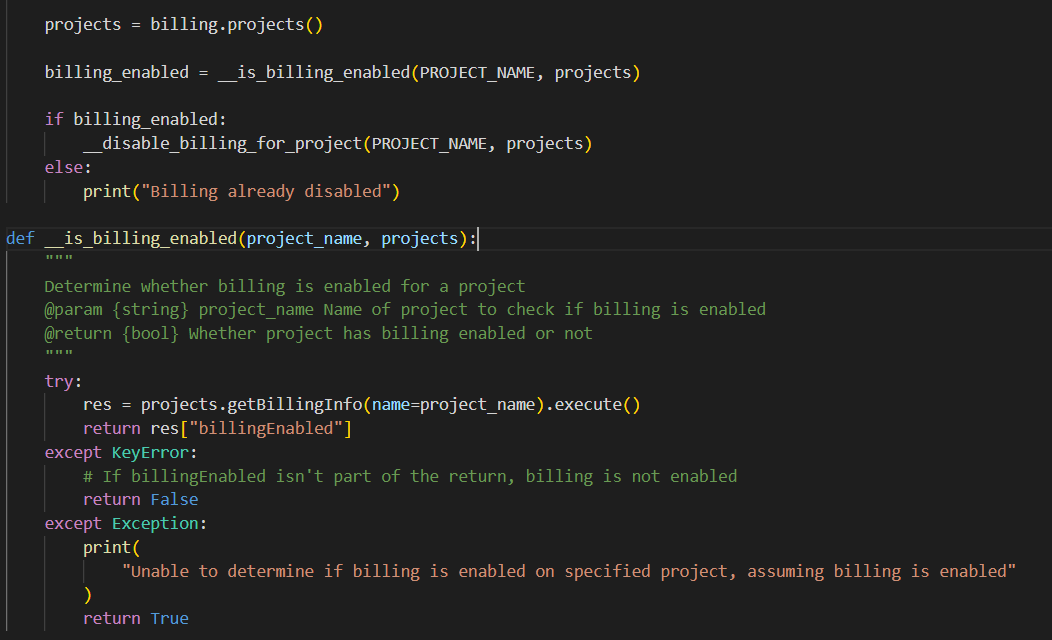


# Budgetabfrage:



Wenn der Pub/Sub Trigger vom Budget ausgelöst wird, wird diese Funktion aufgerufen. Zuerst wird der Base64 Code decodiert und zu einer json-Datei umgewandelt und danach werden die angefallenen Kosten sowie die Budgethöhe abgefragt. Wenn die angefallenen Kosten niedriger sind als das Budget wird ausgegeben das nichts geändert wurde und die aktuelle Höhe der Kosten. Falls die Kosten höher sind, wird die Billing Abfrage aufgerufen.

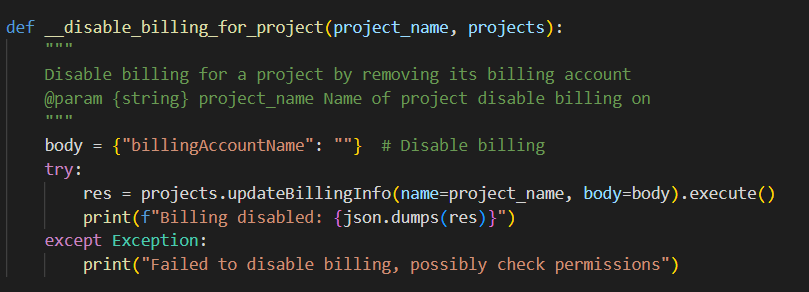
Billing Abfrage:



Falls die angefallenen Kosten höher sind als das Budget wird geguckt, ob das Billing für dieses Projekt noch aktiviert ist. Wenn bei der Methode „\_\_is\_billing\_enabled“ True ausgibt wird die Methode „\_\_disable\_billing\_for\_projekt“ ausgeführt und für das angegebene Projekt wird das Billing deaktiviert.

Die Funktionsweise von „\_\_is\_billing\_enabled“ besteht darin die Billing-Information von dem jeweiligen Projekt abzufragen und falls keine Information ausgegeben wird, ein False und eine Nachricht zu senden das, dass Billing schon deaktiviert wurde.

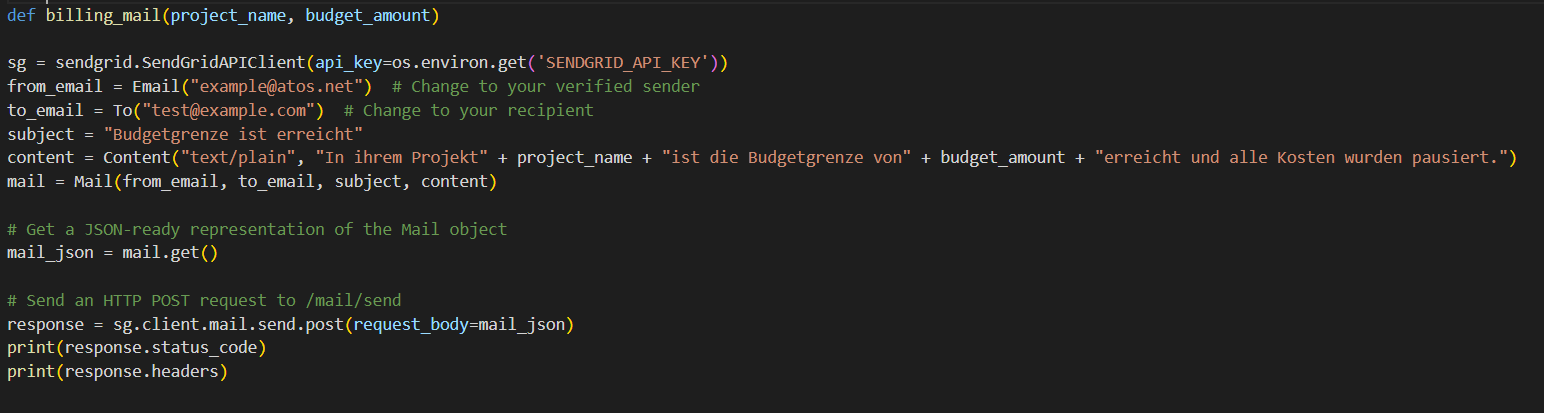
# Disable Billing:



Die Methode ist der Hauptbestandteil dieses Projektes. Sie deaktiviert das Billing für die jeweiligen Budgets, indem sie den Billing Account aus dem angegeben Projekt entfernt.

Für mehr Informationen: <https://cloud.google.com/billing/docs/how-to/notify?utm_source=youtube&utm_medium=unpaidsocial&utm_campaign=mir-20191022-Programmatic-Budget-Notifications&hl=de#functions_billing_stop-python>

Alert Mail:



Die Alert-Mail wird via SendGrid versendet und beinhaltet den Projektnamen sowie die Höhe des Budgets. Man muss, um sich mit der SendGrid API zu verbinden ein Konto erstellen und den API-Key in der ersten Zeile einfügen. Danach gibt man an, von wem die Mail kommt, an wen sie gehen soll, den Betreff und den Inhalt. Diese Mail wird dann als json-Datei versendet und mithilfe von „response.status\_code“ überprüft ob sie angekommen ist.

Für mehr Informationen: <https://docs.sendgrid.com/for-developers/sending-email/quickstart-python>