Google Audit Trigger Documentation

A Zenphi automation flow can be triggered by a Google Audit events for the following Google applications:

- Access transparency
- Admin
- Google Calendar
- Google Chat
- Google Drive
- Google Cloud
- Google Groups
- Enterprise Groups
- Jamboard
- Login
- Google Meet
- Mobile devices
- Rules
- SAML
- OAuth Token
- User accounts
- Context aware access
- Google Chrome
- Data Studio
- Google Keep
- Google Plus (legacy)

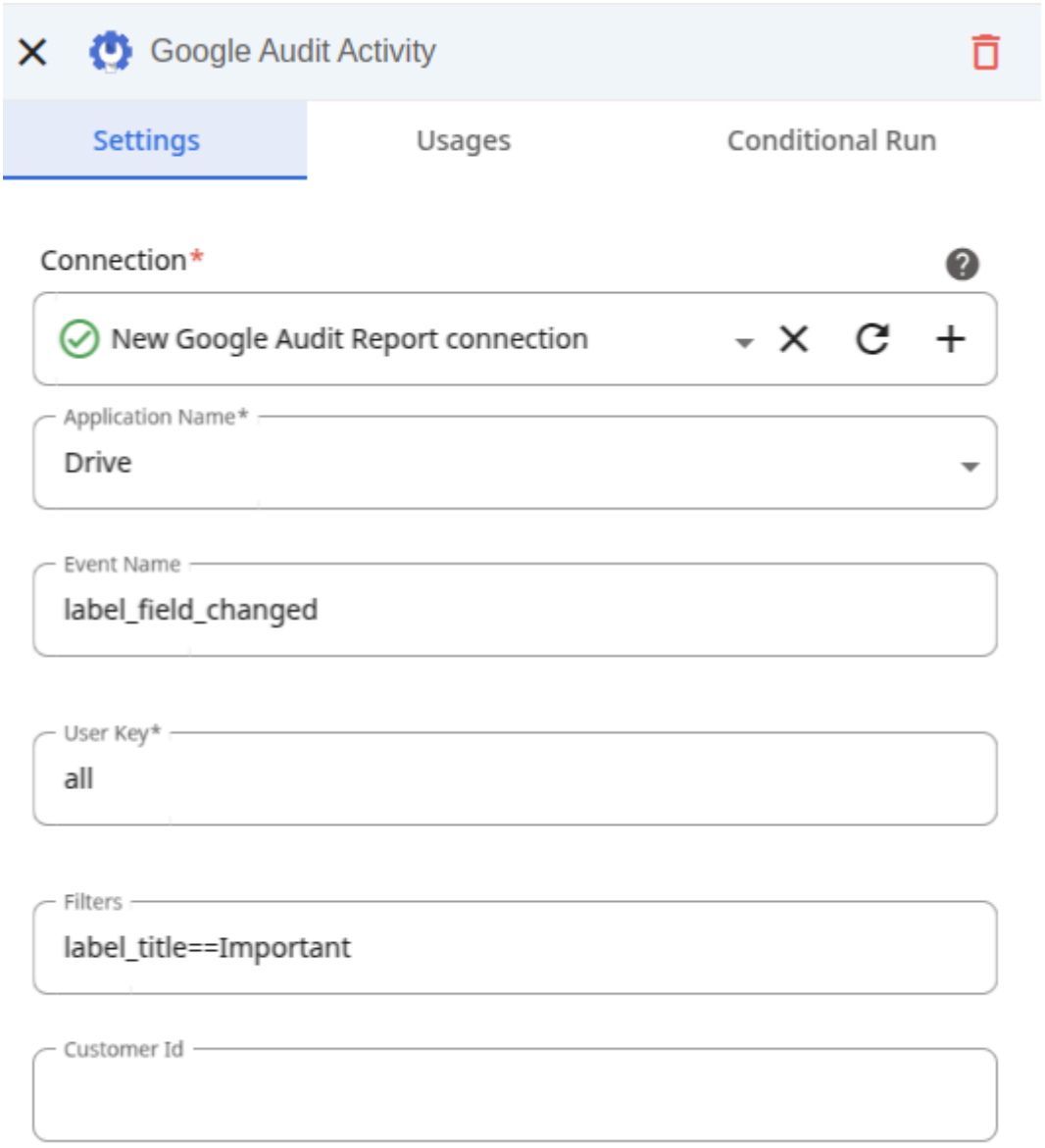
To configure the trigger you have the following fields available:

- **Connection** (required): To set up the trigger you will need to first authenticate the connection with an account with sufficient access.
- **Application Name** (required): Then select the application for which you will be listening to Audit events, which makes part of the Activities REST Resource.
- **Event Name** (optional): Under the documentation for the selected application you will have a number of named events, enter the one you want to work with in the event name field. See example below, using an event from the Drive Audit Activity Events. If you do not set this parameter, the flow will trigger for all activity within the application selected.
- **User Key** (required, *default* = *all*): The user key field accepts either *all* for all users or an *email address* for a specific user account.
- **Filters** (optional): Now you have the option to **filter** down the specific *parameter* of the event you listen to. To find your filter setting, browse the list of parameters available under the event name you set earlier. In the example below I am listening to a name change of a label. When a label is set for an item in Drive, it sets the label_title to the name of the label selected. The labels available on my system have got “Important” defined, and then I listen for that in the filter by simply stating the parameter label_title==Important. Double equal (==) characters is necessary to state

the value for the parameter. *If you do not add anything in this field, it will trigger on all activities within the event type stated in Event Name.*

- **Customer ID** (optional): This parameter is needed if you have Audit access to more than one Google Workspace license group. It refers to the specific ID of a license. *If left empty it will access all audit events available to you within the application selected*.

With the trigger now correctly set up you can start working with the output from the trigger.



The output from the trigger will present the following information:

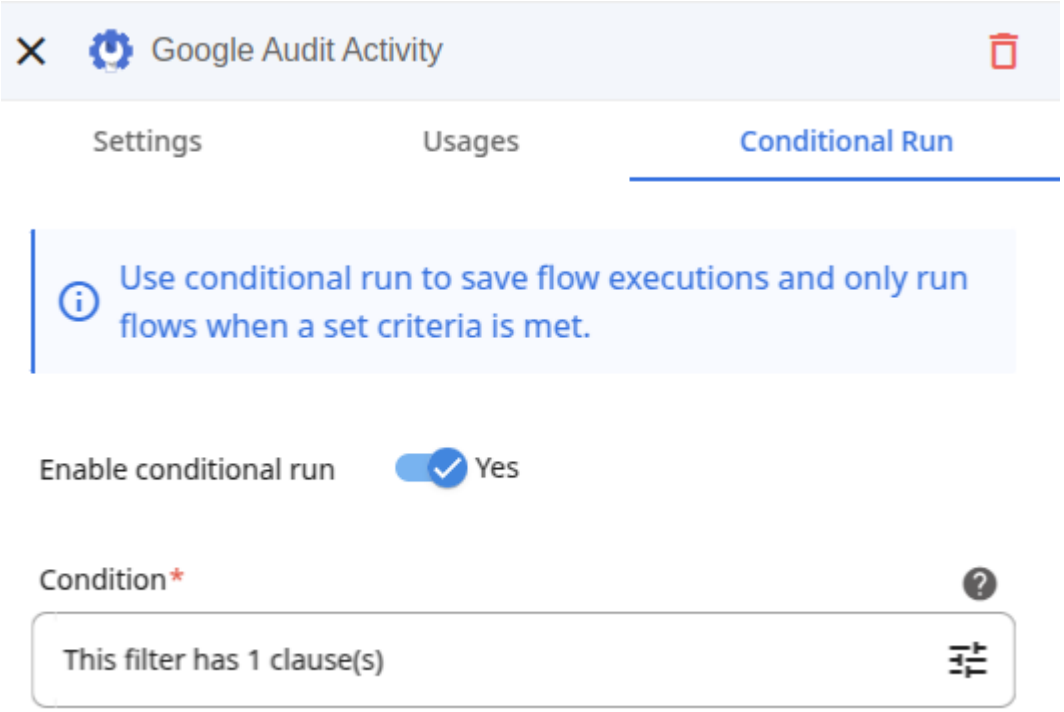
- Id
    - Application Name - Application name to which the event belongs.

    - Event Time - The time of the event.
    - Event Unique Qualifier - Unique qualifier if multiple events have the same time.
    - Customer Id -The unique identifier for a Google Workspace account.
- Actor - Information about the user who has done the activity.
    - Actor Email - The primary email address of the user whose activity is being reported.
    - Actor Type - The type of author who performed the event.
    - Profile Id - The user-unique Google Workspace profile ID.
- Event Name - The name of the event.
- Event Type - Type of event.
- Parameters
    - Name - The name of the parameter.
    - Value - String value of the parameter..
    - Value Type - The data type of the value.
- Owner Domain - The domain that is affected by the event.
- IP Address - IP address of the user doing the action.

In the example here, I would be able to find the Document ID for the document the label was applied to under the output item *Parameters* as:

```
{
    "name": "doc_id",
    "value": "[File ID]",
    "type": "string"
}
```

Like this we can retrieve the document and apply further actions with it in the automation.



We can also apply Conditional Run to the trigger based on values in the Audit event. Below is an example where it will only run if the event was caused by me:

The value of the Actor email could have been filtered in the primary configuration window as well by setting the user key to my email. The conditional run is simply an added layer of trigger rules that can be set up to avoid running flows when not desired.

Due to its versatility, the Audit flow is a preferred way of configuring a granular control of when to run automations.

Mikael Klambro
Egoiste Zenphi Application Consultant