Last update: 9/10/24

WCI: WhatsApp Conversion Import

Site Access: https://github.com/google/wci Video guide: youtu.be/OVXIO5RMHX8

WCI is an open source solution that enables advertisers who offer a WhatsApp channel to measure, attribute, target and optimize their campaigns through signals received in-chat messages. The solution offers a way to integrate and track conversions that happen in business-account WhatsApp chats by linking clicked leads (click to chat) to final conversions (scheduled events, purchased through chat app, etc). As a result, WCI allows advertisers to bring visibility to the WhatsApp journey; measure in-WhatsApp chat interactions; attribute in-WhatsApp chat conversions and target audience-lists with Customer Match.

This is not an officially supported Google product.

Content

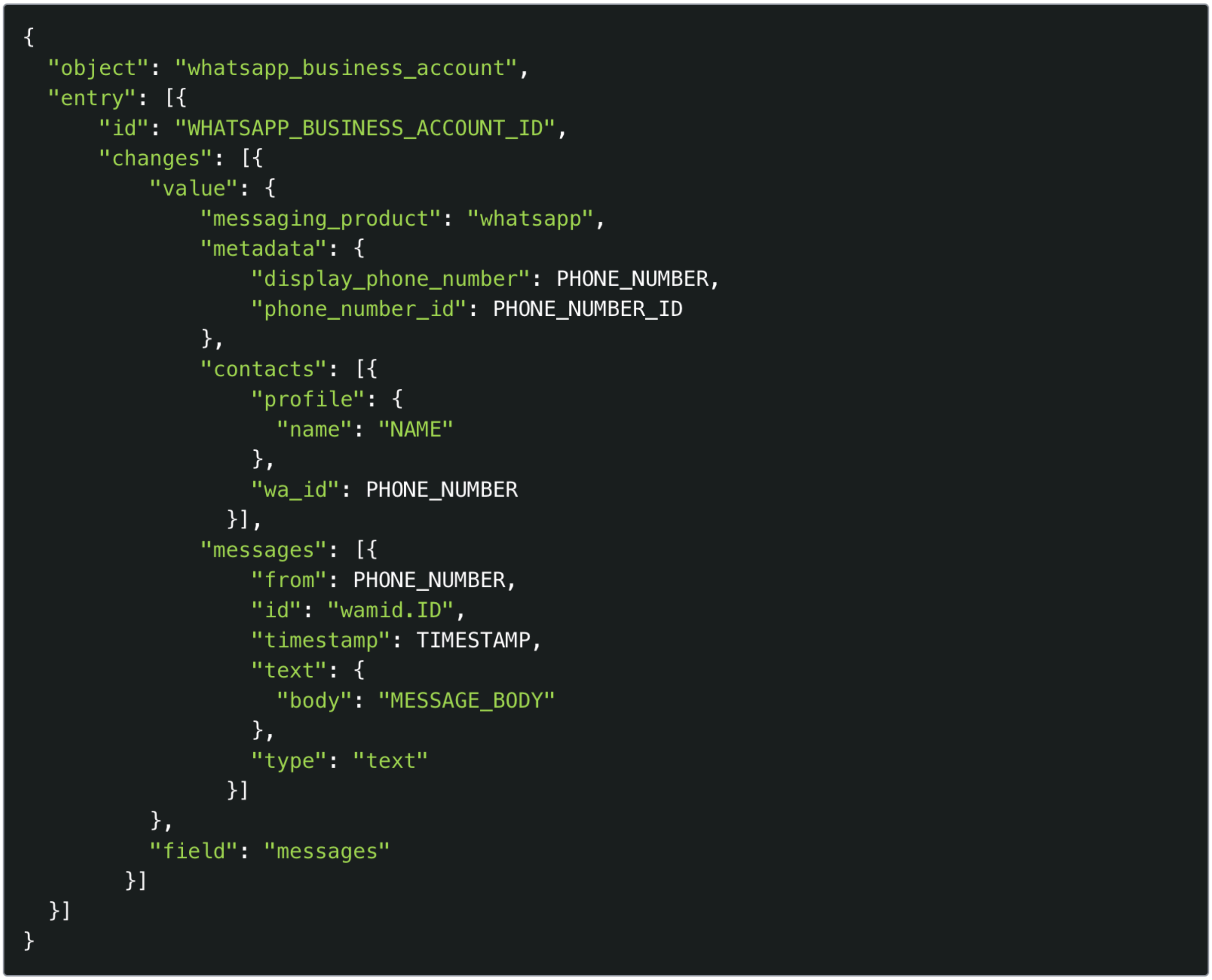
Requirements
APIs
Deployment
  GCP Instructions
  Connect with your WhatsApp Business Account
  Connect with Message Ads
  Container Structure
  Capturing identifiers from website
Workflow
Activation
  Conversions
    Conversion samples
  Customer Match
Resources

Requirements

- Google Cloud Platform account
  - Billing enabled
  - BigQuery enabled
  - Artifact Registry enabled
  - Cloud Run enabled - **endpoints must have public access**
- WhatsApp Business Account
  - One of the following:
    - Direct Setup
      - Facebook developer account
      - Meta App created
      - WhatsApp Business Platform Cloud API enabled
    - Bot Setup
      - Supported Botmaker, Take (Blip), Smaters; or
      - Custom integration between Advertiver's bot with the WCI solution. Refer to the required change to receive the messages from the bot
- Python3
- Google Cloud SDK

APIs

WCI relies on WhatsApp Business Platform Cloud API's webhooks to obtain and process messages received by a WhatsApp Business Account. This process occurs by subscription where every single received message is sent through a request to an externalized WCI endpoint. In that request, its payload has the following format:



```
{
  "object": "whatsapp_business_account",
  "entry": [{
      "id": "WHATSAPP_BUSINESS_ACCOUNT_ID",
      "changes": [{
          "value": {
              "messaging_product": "whatsapp",
              "metadata": {
                  "display_phone_number": PHONE_NUMBER,
                  "phone_number_id": PHONE_NUMBER_ID
              },
              "contacts": [{
                  "profile": {
                    "name": "NAME"
                  },
                  "wa_id": PHONE_NUMBER
                }],
              "messages": [{
                  "from": PHONE_NUMBER,
                  "id": "wamid.ID",
                  "timestamp": TIMESTAMP,
                  "text": {
                    "body": "MESSAGE_BODY"
                  },
                  "type": "text"
                }]
          },
          "field": "messages"
        }]
  }]
}
```

Image 1 - Webhook Notification Payload of Received Messages
Source: https://developers.facebook.com/docs/whatsapp/cloud-api/webhooks/payload-examples

Deployment

The deployment is straightforward and can be initiated with a single command. The solution automates the creation and allocation of Google Cloud resources required for operation:

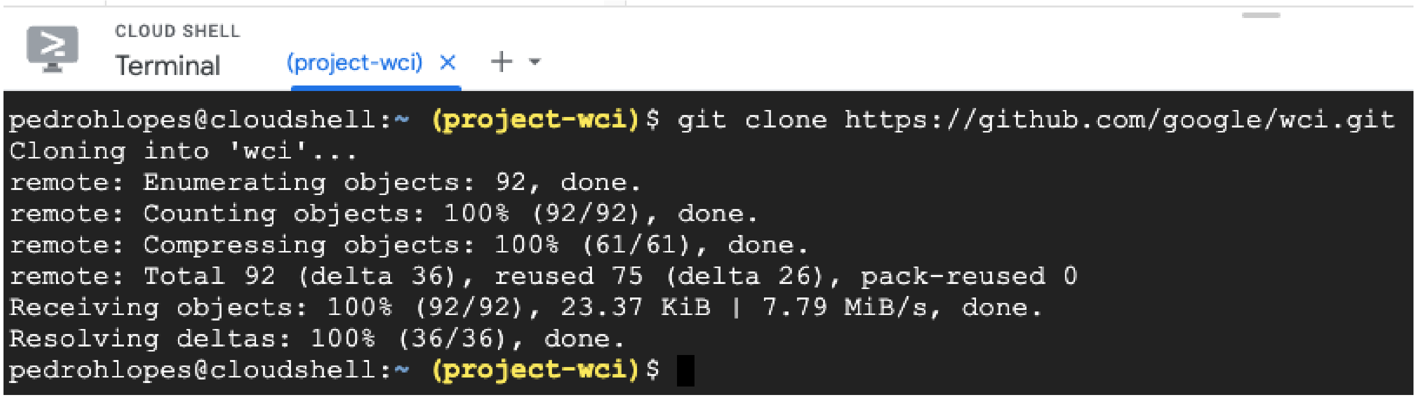
- Service account
- BigQuery dataset and tables
- Artifact Registry

- Cloud Run

GCP Instructions

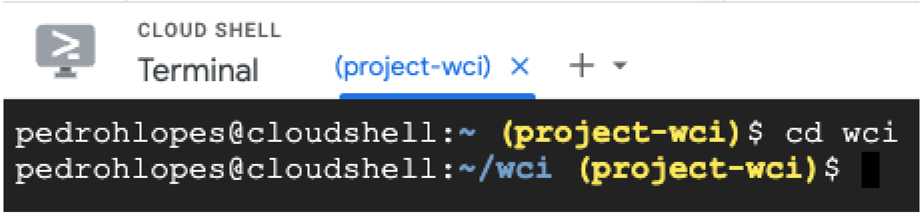
In the Google Cloud's shell, follow the commands below to deploy the solution:

1. Clone the WCI source code from the Github repository with: *git clone https://github.com/google/wci.git*



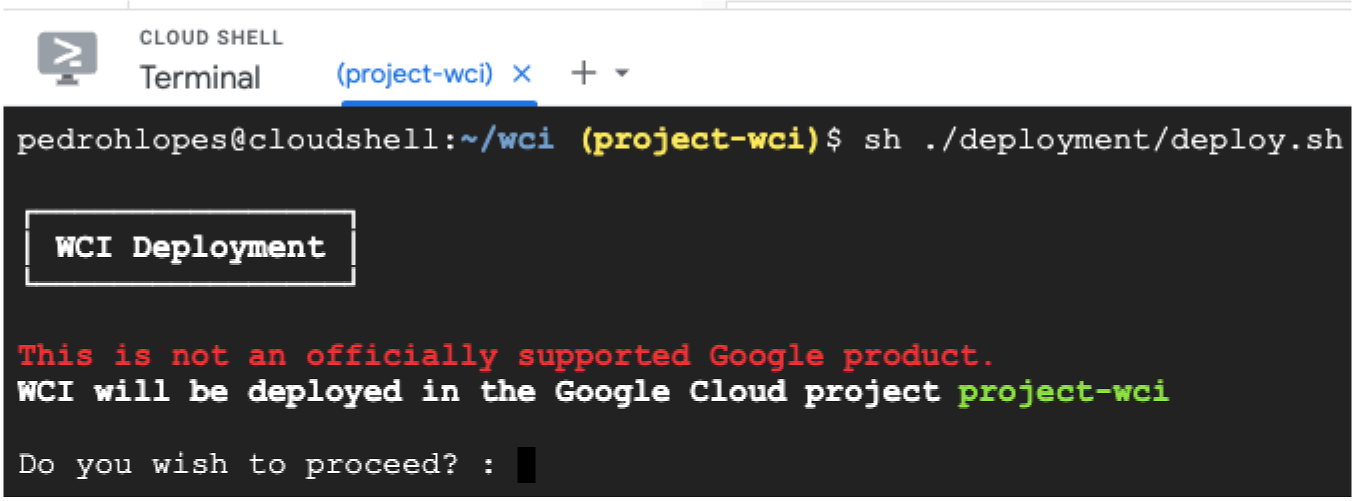
```
pedrohlopes@cloudshell:~ (project-wci)$ git clone https://github.com/google/wci.git
Cloning into 'wci'...
remote: Enumerating objects: 92, done.
remote: Counting objects: 100% (92/92), done.
remote: Compressing objects: 100% (61/61), done.
remote: Total 92 (delta 36), reused 75 (delta 26), pack-reused 0
Receiving objects: 100% (92/92), 23.37 KiB | 7.79 MiB/s, done.
Resolving deltas: 100% (36/36), done.
pedrohlopes@cloudshell:~ (project-wci)$
```

2. The command above creates a folder called wci. Open that folder with: *cd wci*



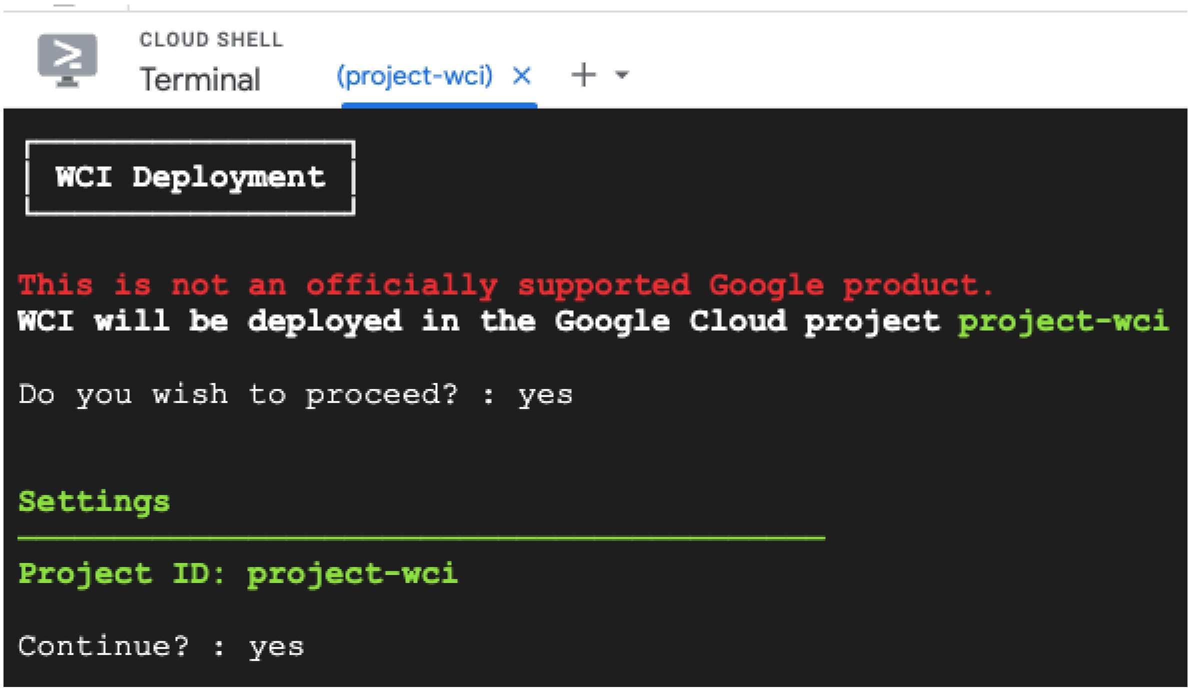
```
pedrohlopes@cloudshell:~ (project-wci)$ cd wci
pedrohlopes@cloudshell:~/wci (project-wci)$
```

3. To start, execute: *sh ./deployment/deploy.sh*



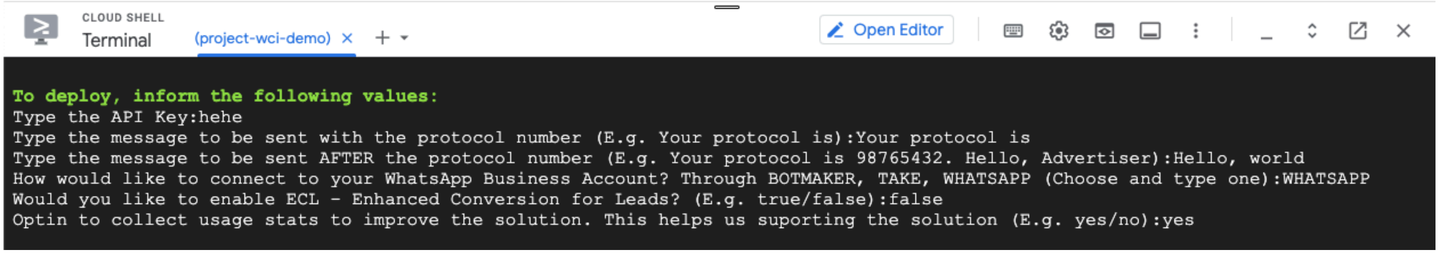
```
pedrohlopes@cloudshell:~/wci (project-wci)$ sh ./deployment/deploy.sh

 ┌─────────────────┐
 │ WCI Deployment  │
 └─────────────────┘

This is not an officially supported Google product.
WCI will be deployed in the Google Cloud project project-wci

Do you wish to proceed? :
```

4. Initiate the deployment confirming the selected project and typing: *yes*



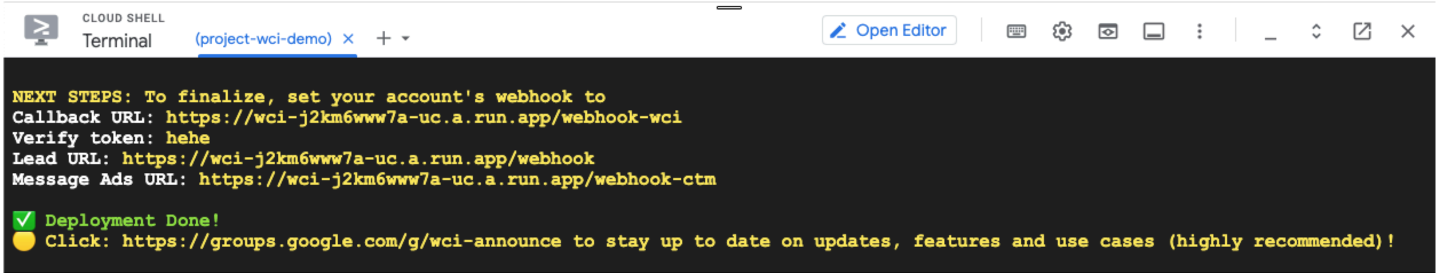
```
WCI Deployment

This is not an officially supported Google product.
WCI will be deployed in the Google Cloud project project-wci

Do you wish to proceed? : yes


Settings
__________________________________________
Project ID: project-wci

Continue? : yes
```

5. Follow and answer the prompts in order to configure the service/



```
To deploy, inform the following values:
Type the API Key:hehe
Type the message to be sent with the protocol number (E.g. Your protocol is):Your protocol is
Type the message to be sent AFTER the protocol number (E.g. Your protocol is 98765432. Hello, Advertiser):Hello, world
How would like to connect to your WhatsApp Business Account? Through BOTMAKER, TAKE, WHATSAPP (Choose and type one):WHATSAPP
Would you like to enable ECL - Enhanced Conversion for Leads? (E.g. true/false):false
Optin to collect usage stats to improve the solution. This helps us suporting the solution (E.g. yes/no):yes
```

6. With the App deployed, the script prints the Lead URL endpoint that should be used to capture the gclids before redirecting to WhatsApp.

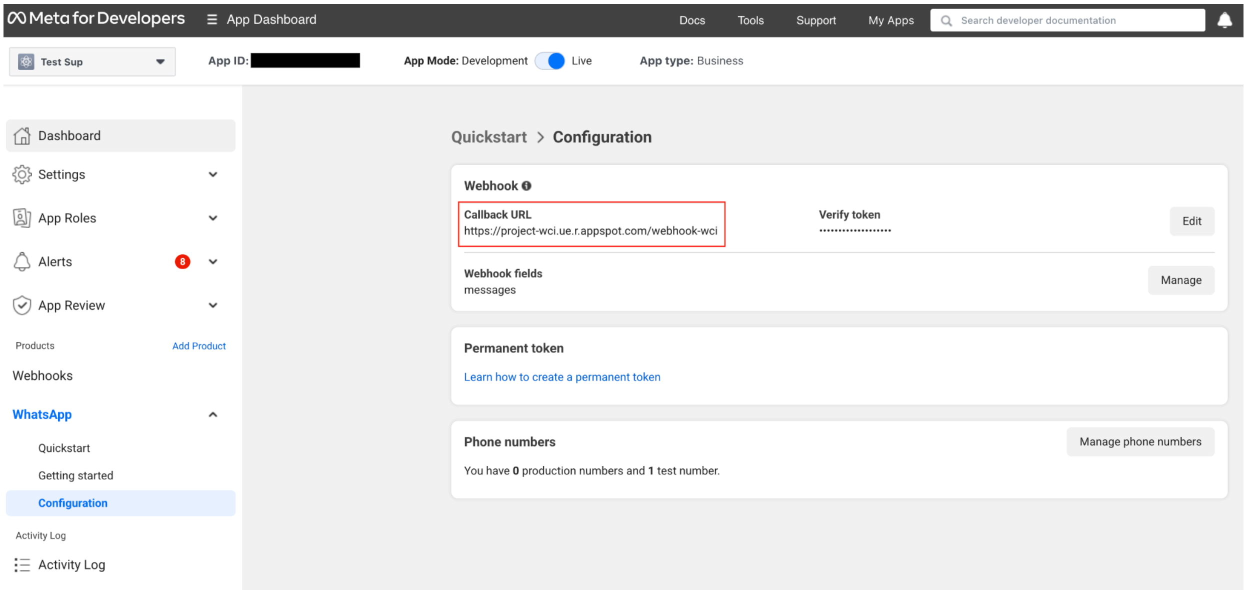


```
NEXT STEPS: To finalize, set your account's webhook to
Callback URL: https://wci-j2km6www7a-uc.a.run.app/webhook-wci
Verify token: hehe
Lead URL: https://wci-j2km6www7a-uc.a.run.app/webhook
Message Ads URL: https://wci-j2km6www7a-uc.a.run.app/webhook-ctm

✅ Deployment Done!
🟡 Click: https://groups.google.com/g/wci-announce to stay up to date on updates, features and use cases (highly recommended)!
```

Connect with your WhatsApp Business Account

7. WCI supports connection with WhatsApp, Botmaker or Take (Blip) via webhook. This choice is made during the deployment of the solution.

   a. For WhatsApp direct connection, configure the Callback URL endpoint at your App's configuration page. More information at [WhatsApp Business Platform Cloud API's webhooks]().



App ID: [redacted]

App Mode: Development Live

App type: Business

Callback URL
https://project-wci.ue.r.appspot.com/webhook-wci

Verify token
••••••••••••••••••••

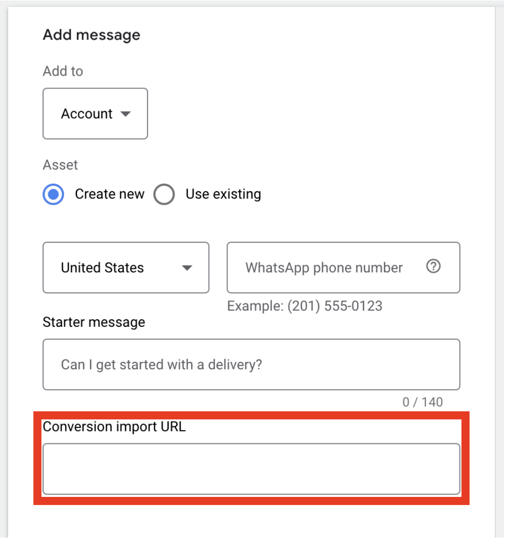
Webhook fields
messages

You have **0** production numbers and **1** test number.

   b. For the others, inform the Callback URL to your bot representative in order to setup the webhook with your provider

[CLOSED BETA] Connect with Message Ads

8. WCI can be leveraged with Message Ads. To get started, use the Message Ads URL in your Message Assets in Google Ads:



Add message

Add to

Account

Asset

● Create new ○ Use existing

United States

WhatsApp phone number

Example: (201) 555-0123

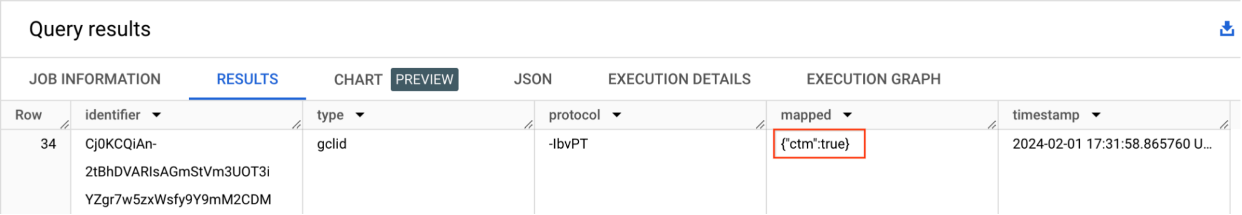
Starter message

Can I get started with a delivery?

0 / 140

Conversion import URL

a. Once set up, the Conversion Import URL will submit a POST into WCI passing chat_id as protocol number and gclid for every single click on the Message Ads extension. As a result, a new row will be added to the pending_leads table for every click.



Query results

Row	identifier	type	protocol	mapped	timestamp
34	Cj0KCQiAn-2tBhDVARIsAGmStVm3UOT3iYZgr7w5zxWsfy9Y9mM2CDM	gclid	-IbvPT	{"ctm":true}	2024-02-01 17:31:58.865760 U...

,

Container Structure

Once the deployment is successfully completed, the following structure will be in place:

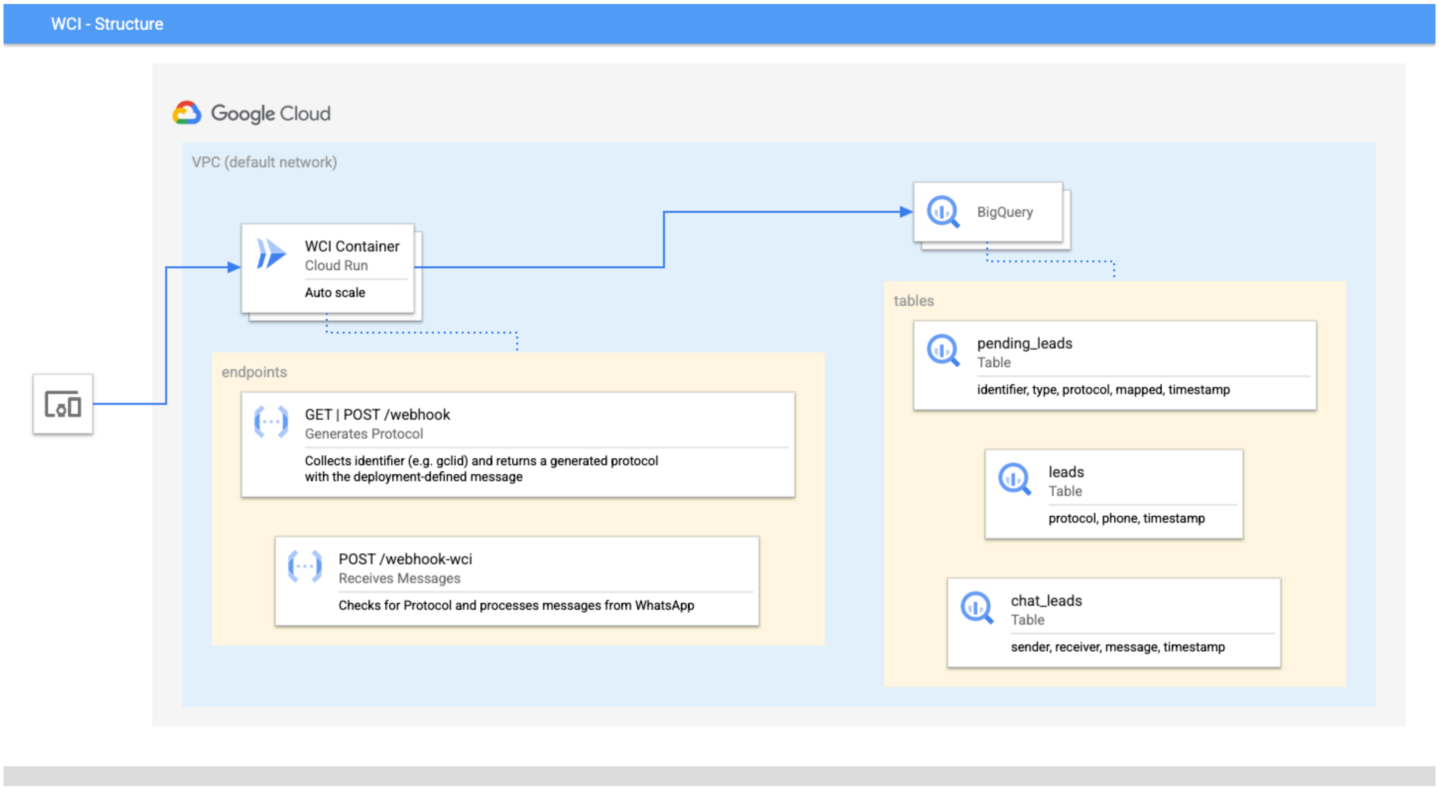

WCI - Structure
Google Cloud
VPC (default network)
WCI Container
Cloud Run
Auto scale
BigQuery
endpoints
GET | POST /webhook
Generates Protocol
Collects identifier (e.g. gclid) and returns a generated protocol with the deployment-defined message
POST /webhook-wci
Receives Messages
Checks for Protocol and processes messages from WhatsApp
tables
pending_leads
Table
identifier, type, protocol, mapped, timestamp
leads
Table
protocol, phone, timestamp
chat_leads
Table
sender, receiver, message, timestamp

Image 2 - WCI Resources Diagram

Capturing identifiers from website

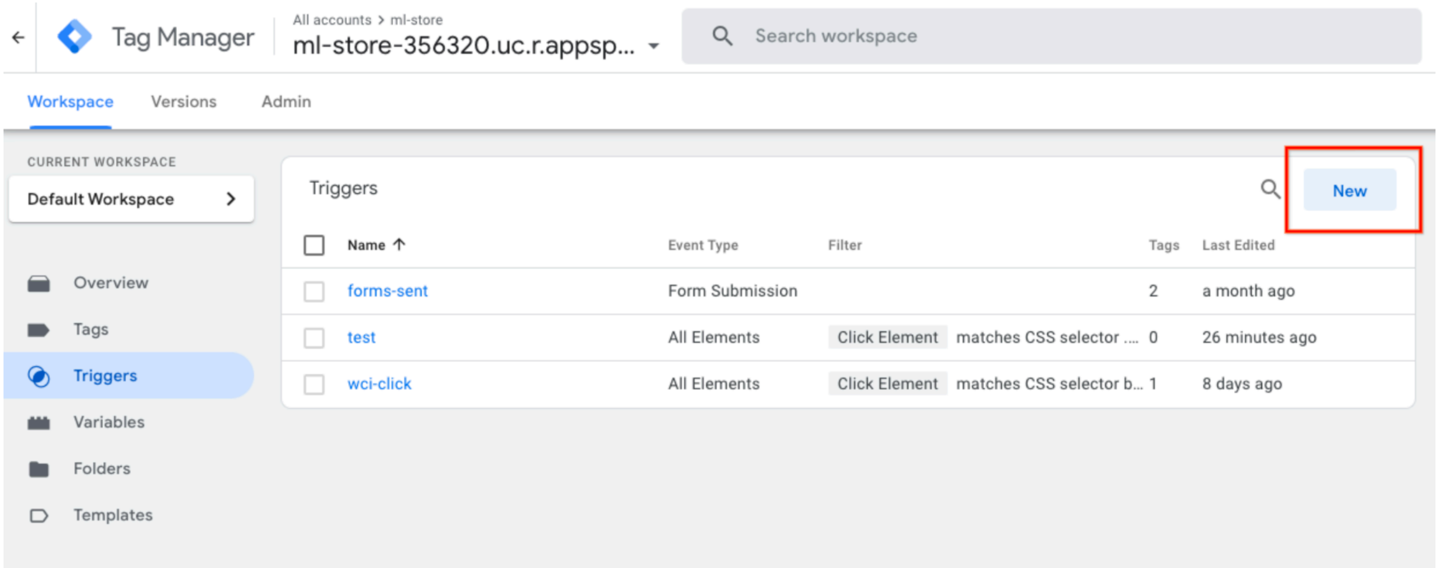
Lastly, to start capturing the leads to WhatsApp, the lead endpoint should be used instead of redirecting users directly to WhatsApp. There are two ways to accomplish that: editing your landing page's javascript to use the new target [Lead] URL or using GTM to do so.

- Using the web page's Javascript:

```
JavaScript
<script>
/**  FILL THE VALUES BELOW  */
var CONVERSION_IMPORT_URL = "";  // fill with your servoce URL
var WHATSAPP_NUMBER = "";        // fill with your whatsapp number account. E.g.
+15550801213
var STARTER_MESSAGE = "";        // fill with your initial message. E.g. Hi there

/** DO NOT CHANGE BELOW */
```

```
function encodeMessage(message) {
  return encodeURIComponent(message);
}
function getMessageWithProtocol(protocol_number) {
  return "[Chat ID: " + protocol_number + "] " + STARTER_MESSAGE;
}
function getURL(message) {
  return "https://wa.me/" + WHATSAPP_NUMBER + "?text=" + message;
}
function getProtocolNumber() {
  return crypto.randomUUID()
    .split("")
    .filter(function (value, index, self) {
      return self.indexOf(value) === index;
    })
    .join("")
    .slice(-6);
}
function getGclid() {
  var params = new URLSearchParams(location.search);
  var gclid = params.get("gclid");

  if (!gclid) {
    // Checks for a cookie
    var gcookie = document.cookie.split(";").map(function (cookie) {
      return cookie.split("=");
    });
    gcookie = gcookie.find(function (cookie) {
      return cookie[0].trim() == "_gcl_aw";
    });

    if (gcookie) gclid = gcookie[1].split(".")[2];
  }
  return gclid;
}
function sendToConversionImportUrl(protocol_number, gclid){
  var payload = {};
  var url = new URL(CONVERSION_IMPORT_URL);
  var params = new URLSearchParams(url);
  params.set("type", "gclid");
  params.set("id", gclid);

  fetch(CONVERSION_IMPORT_URL + "/webhook?" + params.toString(), {
    method: "POST",
    headers: {
      "Content-type": "application/json",
    },
    body: JSON.stringify(payload),
  })
    .then(function (response) {
      return response.json();
    })
    .catch(function (err) {
      console.error(err);
    })
    .then(function (data) {
        var message = '';
        if( data && data.message) message = data.message;
        // Replace YOUR_WHATSAPP_NUMBER with yout Whatsapp account number
        location.replace(getURL(message));
    });
 }

var protocol_number = getProtocolNumber();
var message = encodeMessage(getMessageWithProtocol(protocol_number));
var gclid = getGclid()

// Uncomment the folllowing lines if you'd like to have the protocol number
// in the datalyer
// dataLayer.push({
//   'protocol_number': protocol_number,
//   'event': 'click_on_wci'
// });

// If there isn't a gclid, skip the step of passing gclid
// to the webhook
if(!gclid)
   location.replace(location.replace(getURL("")));
else
   sendToConversionImportUrl(protocol_number, gclid);

</script>
```

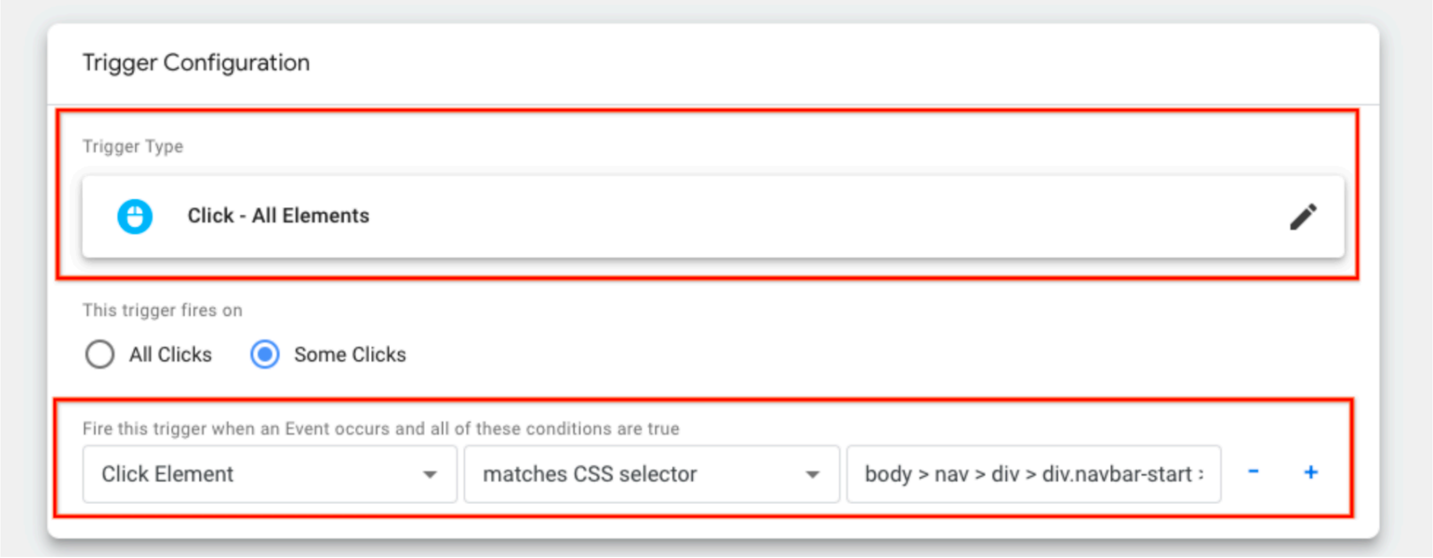
- Using GTM:
  1. Creating the trigger
     1. In the triggers' section, click at *New* to create a new trigger



Triggers

□	Name ↑	Event Type	Filter	Tags	Last Edited
□	forms-sent	Form Submission		2	a month ago
□	test	All Elements	Click Element matches CSS selector	0	26 minutes ago
□	wci-click	All Elements	Click Element matches CSS selector b...	1	8 days ago

2. Next, select the type "Click - All Elements" to limit for a few elements by entering the CSS selector of choice



Trigger Configuration

Trigger Type

Click - All Elements

This trigger fires on

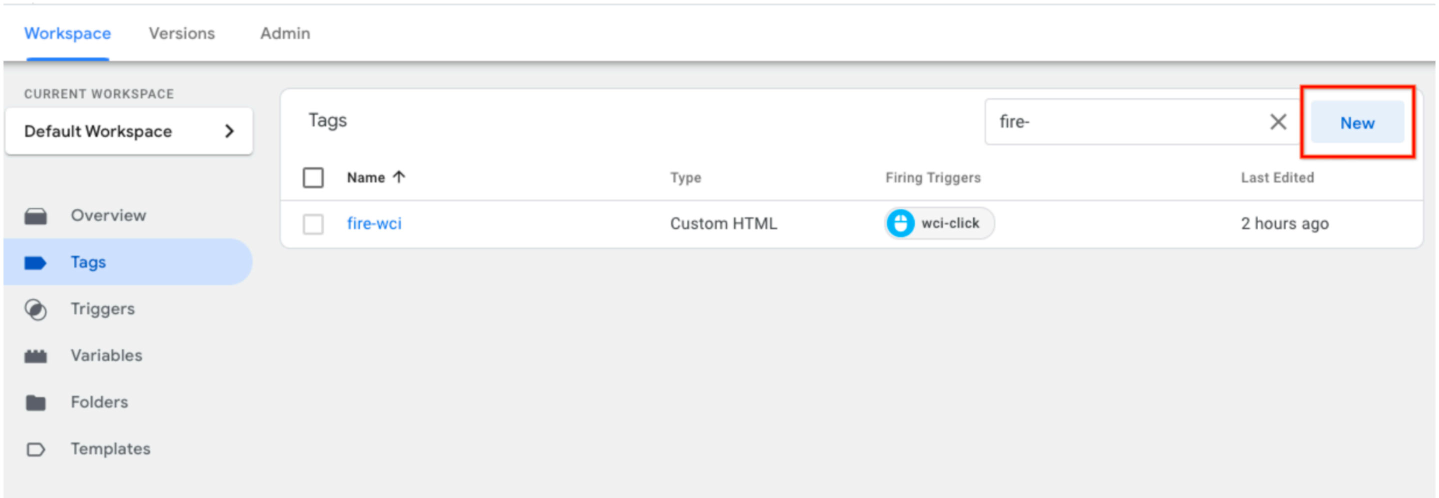
○ All Clicks ● Some Clicks

Fire this trigger when an Event occurs and all of these conditions are true

Click Element | matches CSS selector | body > nav > div > div.navbar-start :

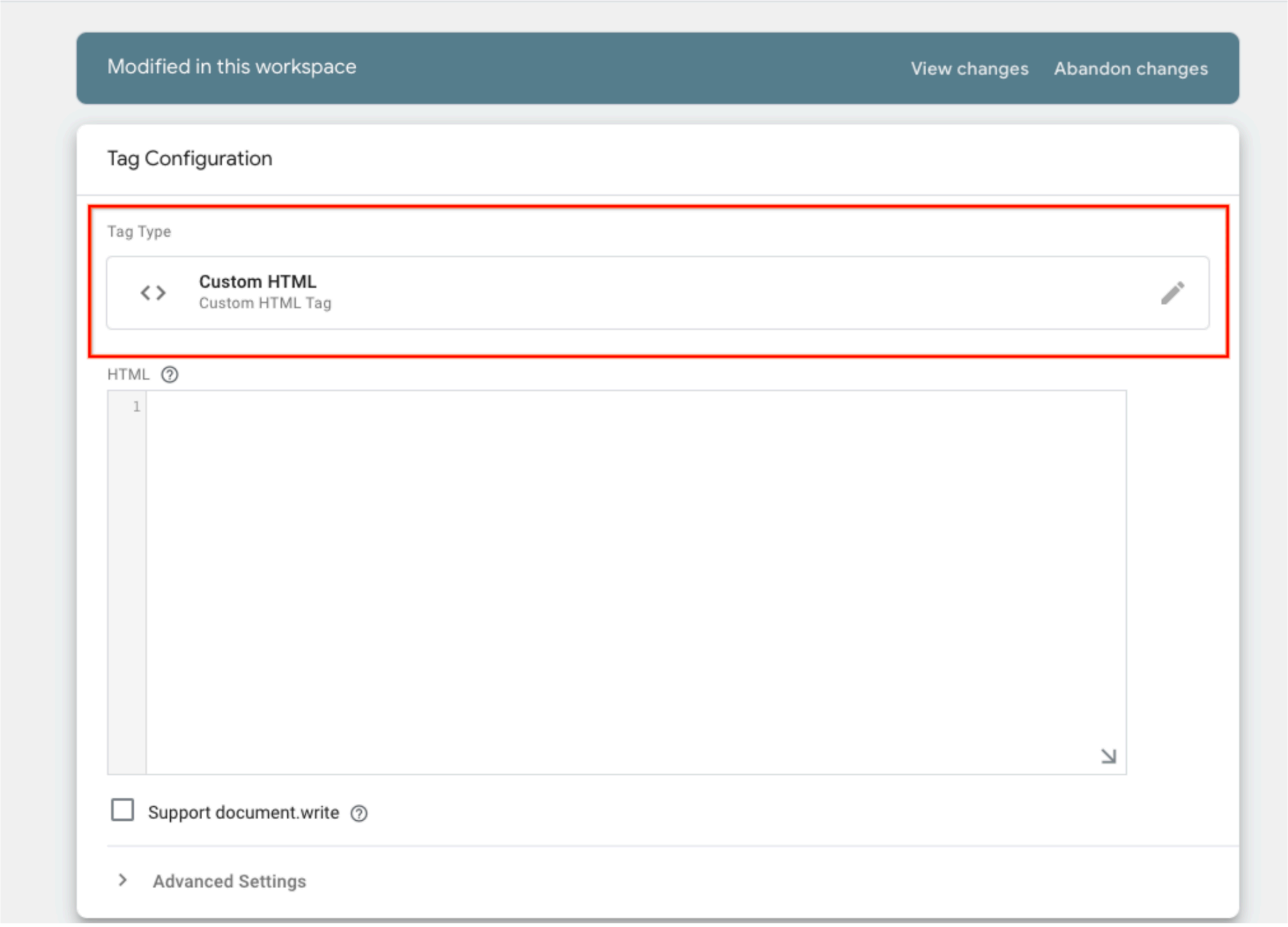
3. Click *Save*

2. Adding the Tag
   1. In the Tag's section, click at *New* to create a new tag



Name ↑	Type	Firing Triggers	Last Edited
fire-wci	Custom HTML	wci-click	2 hours ago

2. Next, select "Custom HTML"

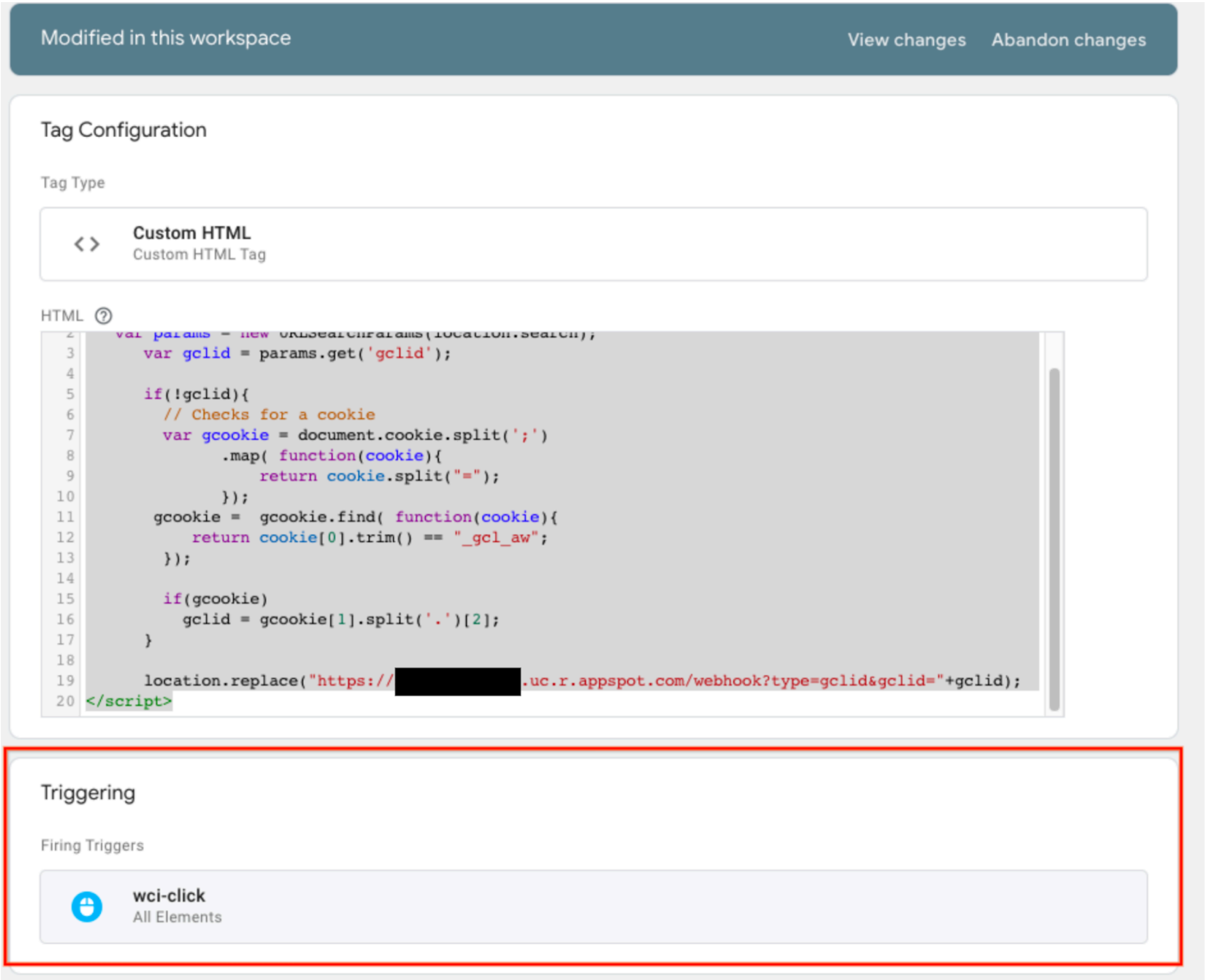


Tag Type
Custom HTML
Custom HTML Tag

3. Snippet
   1. Include the snippet below in the HTML tag area

```
JavaScript
<script>
   var params = new URLSearchParams(location.search);
      var gclid = params.get('gclid');

      if(!gclid){
        // Checks for a cookie
        var gcookie = document.cookie.split(';')
              .map( function(cookie){
                  return cookie.split("=");
              });
       gcookie =  gcookie.find( function(cookie){
           return cookie[0].trim() == "_gcl_aw";
        });

        if(gcookie)
          gclid = gcookie[1].split('.')[2];
      }

      // If there's any additional value to associate with the click,
      // include inside the payload in order to be saved into the
      // mapped column on the table
      var payload = {};

       // Replace YOUR_ENDPOINT with the URL of the WCI service
      fetch("https://{{YOUR_ENDPOINT}}/webhook?type=gclid&id="+gclid,{
         method: 'POST',
         headers: {
           'Content-type':'application/json'
         },
         body: JSON.stringify(payload)

      })
      .then(function(response){
        return response.json();
      })
      .catch(function(err){
        console.error(err);
      })
      .then(function(data){
        var message = '';
        if( data && data.message) message = data.message;
        // Replace YOUR_WHATSAPP_NUMBER with yout Whatsapp account number
        location.replace("https://wa.me/+{{YOUR_WHATSAPP_NUMBER}}?text="+message);
      });

</script>
```

2. Add the trigger created above to trigger the tag being created



Tag Configuration

Tag Type

Custom HTML
Custom HTML Tag

HTML

```
   var params = new URLSearchParams(location.search);
      var gclid = params.get('gclid');

      if(!gclid){
        // Checks for a cookie
        var gcookie = document.cookie.split(';')
              .map( function(cookie){
                  return cookie.split("=");
              });
       gcookie =  gcookie.find( function(cookie){
           return cookie[0].trim() == "_gcl_aw";
        });

        if(gcookie)
          gclid = gcookie[1].split('.')[2];
      }

      location.replace("https://[redacted].uc.r.appspot.com/webhook?type=gclid&gclid="+gclid);
</script>
```

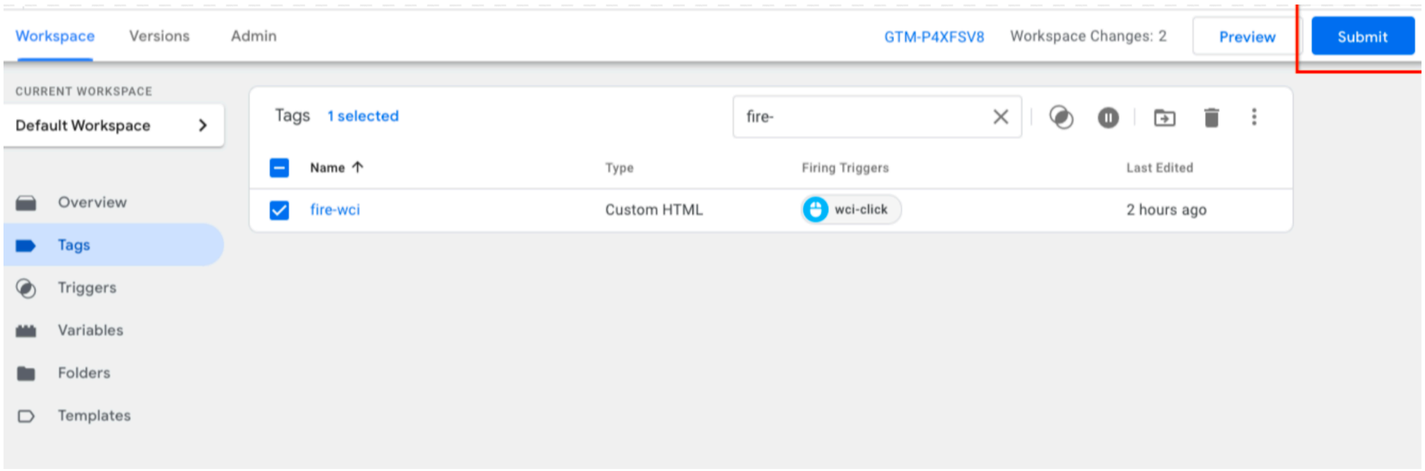
Triggering

Firing Triggers

wci-click
All Elements

3. Click *Save*

4. Publishing the changes
   1. Lastly, submit the changes to apply the workspace changes into production and, consequently, capture WhatsApp leads



Name ↑	Type	Firing Triggers	Last Edited
fire-wci	Custom HTML	wci-click	2 hours ago

Workflow

WCI produces two externalized endpoints and uses a few tables to manage leads and interactions in WhatsApp chats. The first endpoint, named Lead, is responsible for capturing identifiers such as gclids, generating unique protocol numbers and redirecting the request to the deploy-configured WhatsApp Business account chat. The second endpoint, named callback, is in charge of handling the WhatsApp Business Platform Cloud API's webhook's requests and verifying the content of the received messages to check for protocol numbers in order to link leads with senders. The captured (pending) leads, linked leads and in-chat interactions are stored and available in the following tables:

- **pending-leads:** Captures and stores every single generated lead that migrated from the Online to the WhatsApp journey. To put it simply, when the user clicks at the WhatsApp button in the advertiser's webpage, the session is redirected to the Lead endpoint passing an identifier such as gclid through parameter and generating a protocol number.

- **leads:** links identifier (e.g. gclid) > protocol numbers > senders. A lead is created when the user sends the pre-typed message that was generated through the lead endpoint with the protocol number.

- **chat_leads:** Tracks interactions throughout the WhatsApp journey, enabling, for instance, advertisers to identify when users convert (scheduling an event, purchasing, etc).

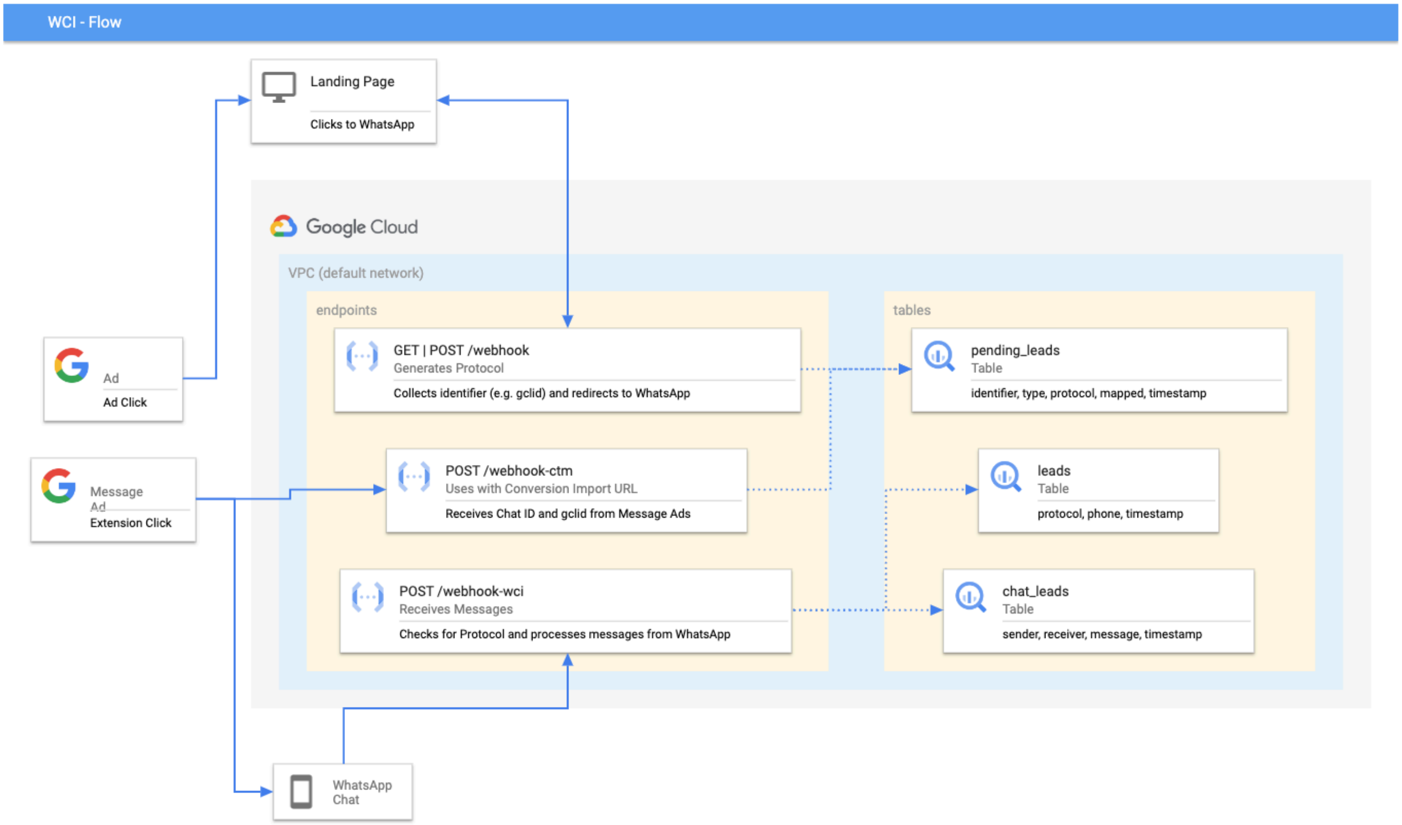
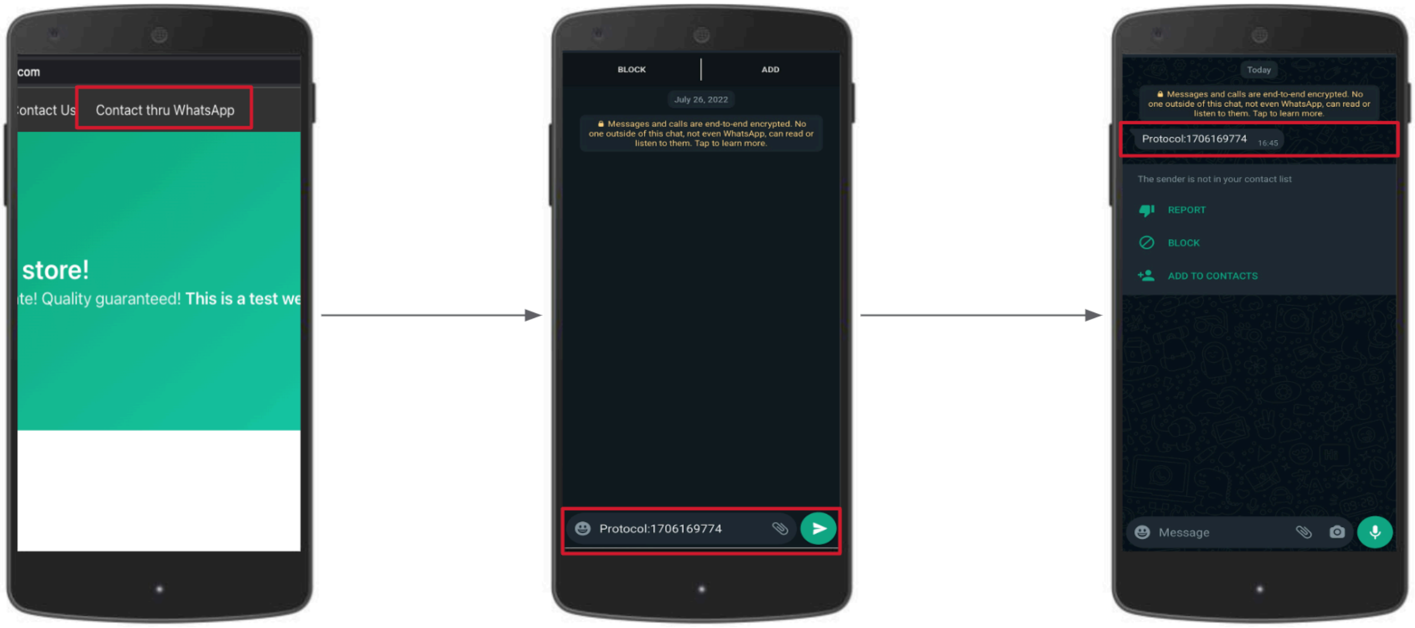

WCI - Flow
Landing Page
Clicks to WhatsApp
Google Cloud
VPC (default network)
endpoints
GET | POST /webhook
Generates Protocol
Collects identifier (e.g. gclid) and redirects to WhatsApp
POST /webhook-ctm
Uses with Conversion Import URL
Receives Chat ID and gclid from Message Ads
POST /webhook-wci
Receives Messages
Checks for Protocol and processes messages from WhatsApp
tables
pending_leads
Table
identifier, type, protocol, mapped, timestamp
leads
Table
protocol, phone, timestamp
chat_leads
Table
sender, receiver, message, timestamp
Ad
Ad Click
Message Ad
Extension Click
WhatsApp Chat

Image 3 - WCI Workflow



Messages and calls are end-to-end encrypted. No one outside of this chat, not even WhatsApp, can read or listen to them. Tap to learn more.

Protocol:1706169774

Messages and calls are end-to-end encrypted. No one outside of this chat, not even WhatsApp, can read or listen to them. Tap to learn more.

Protocol:1706169774 16:45

The sender is not in your contact list

Image 4 - User clicks to WhatsApp >> gClid collected, protocol generated with message pre-filled >> Sender is associated with protocol and gclid

Activation

Conversions

With leads linked to senders, WCI qualifies advertisers to upload in-chat conversions to Google Ads products. The image below illustrates how one could use and target certain messages received through the WhatsApp channel to flag and upload conversions to Google Ads, for example. In addition, it's also possible to use Enhanced Conversions for Leads with the senders' information.

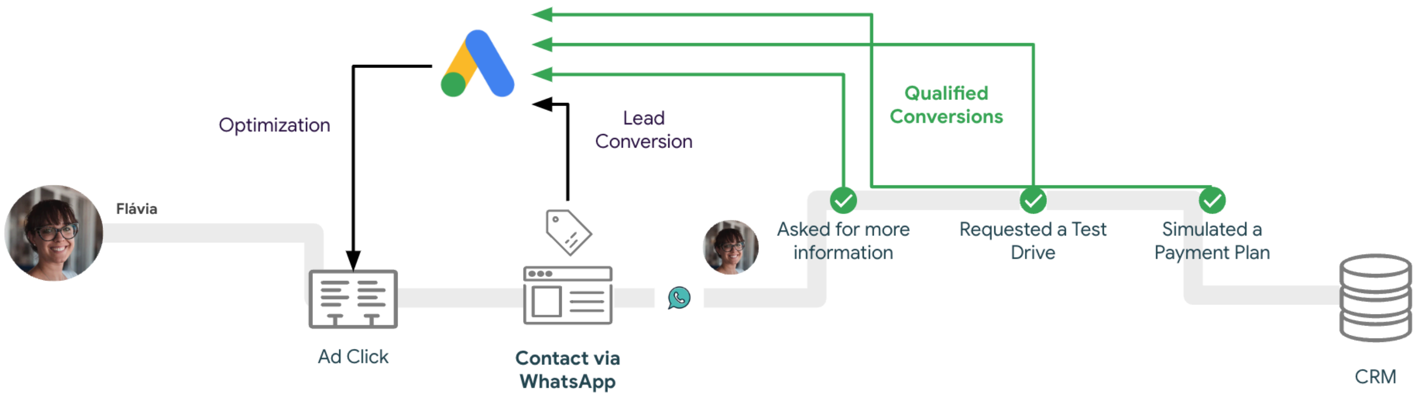


Image 5 - Sample of how to use conversions captured by WCI

Conversion samples

- **First-messages sent:** Retrieve gclids from users that migrated from online to WhatsApp and engaged by sending a message

```
Unset
SELECT
 pl.identifier AS gclid,
 FORMAT_DATETIME('%Y-%m-%dT%H:%M:%S.000', ld.timestamp, 'America/Sao_Paulo') AS time,
 '1' AS amount
FROM
 `{{YOUR_GCP_PROJECT}}.wci.leads` ld
INNER JOIN
 `{{YOUR_GCP_PROJECT}}.wci.pending_leads` pl
USING
 (protocol)
```

```
WHERE
 IFNULL(pl.identifier,'null') <> 'null'
 AND pl.type = 'gclid'
```

- **Users signalized an action:** Retrieve gclids from users that migrated from online to WhatsApp and signalized an action. To illustrate, users that sent “Lets Schedule”

Unset

```
SELECT
 pl.identifier AS gclid,
 FORMAT_DATETIME('%Y-%m-%dT%H:%M:%S.000', ld.timestamp, 'America/Sao_Paulo') AS time,
 '1' AS amount
FROM
 `{{YOUR_GCP_PROJECT}}.wci.leads` ld
INNER JOIN
 `{{YOUR_GCP_PROJECT}}.wci.pending_leads` pl
USING
 (protocol)
INNER JOIN
  `{{YOUR_GCP_PROJECT}}.wci.chat_leads` cl
ON ld.phone = cl.sender # in case the user sent the message
#ON ld.phone = cl.receiver # in case the user received the message
WHERE
  IFNULL(pl.identifier,'null') <> 'null'
  AND pl.type = 'gclid'
  AND lower(cl.message) LIKE 'lets schedule' #adjust it accordingly to your business needs
```

.

Customer Match

Since senders’ information is available through WhatsApp Business Platform Cloud API, advertisers are able to leverage that information to upload and target audience lists. For example, creating and using audience lists for all users that did not finalize a purchase. Moreover, depending on the chosen identifier, other ways of Customer Match could also be used - e.g. user/device ids.

FAQs

Q: Is the Chat ID required?

A: Yes, it is. The sole purpose of the Chat ID is to bridge the connection between online and offline (on WhatsApp) experiences.

Q: What happens if the user deletes the Chat ID before sending the message?
A: Users have the freedom to modify or delete the chat ID. However, during our testing, we observed that about 70% of users do not change or delete the initial message, and those who do not change the pre-typed message are more likely to convert. Therefore, it can be regarded as a "natural qualifying" acquisition process.

Q: Does Google have access to the content of the messages?
A: No, Google does not access any information exchanged in chats. However, the client utilizing the infrastructure may decide to share a piece of information, the Google Click ID (GCLID), with their Google Ads account.

Q: Is it necessary to store the GCLID (Google Click ID)?
A: Yes, storing the GCLID is crucial for measuring, attributing, and optimizing campaigns by uploading GCLIDs back into Google Ads.

Resources

Deployment video-guide
https://youtu.be/OVXIO5RMHX8

WCI's source code
https://github.com/google/wci

WhatsApp Business Platform Cloud API
https://developers.facebook.com/docs/whatsapp/cloud-api/guides/set-up-webhooks

Google Cloud's Artifacts Registry
https://cloud.google.com/artifact-registry/docs

Google Cloud's Run
https://cloud.google.com/run/docs

Google Cloud's BigQuery
https://cloud.google.com/bigquery/docs

Google Ads' Conversion API
https://developers.google.com/google-ads/api/docs/conversions/overview

Google Ads' Customer Match API

https://developers.google.com/google-ads/api/docs/remarketing/audience-types/customer-match

Google Ads’ Enhanced Conversion for Leads API
https://developers.google.com/google-ads/api/docs/conversions/upload-identifiers