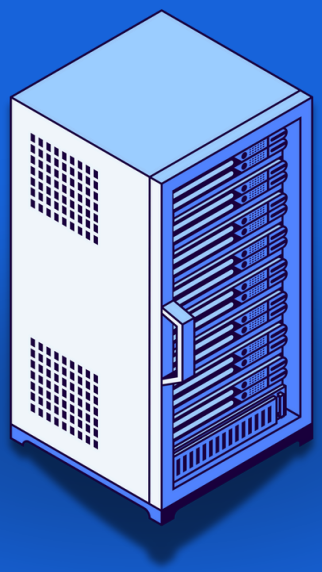
ANALYGO

Server-Side Tracking



HAMZA EL KHARRAZ

The Agency Guide to a Fast and Successful Server-Side Setup.

Hi, and thank you for downloading this guide. I'm Hamza, the guy behind ANALYGO, a data marketing consultancy that works mainly with Ad and media agencies.

If you have any questions or want to reach out, feel free to use this email address: hello@analygo.co

If you are interested in working with me on a long-term project, scan the QR code below or head to the contact page.

SCAN ME

Table Of Contents

Introduction

This guide is not meant to be a technical manual (although it does mention some fundamental technical concepts), but it can serve as a solid introduction to the world of server-side tracking. If you are well-versed in the fundamentals, this book won't be of much help to you, aside from the last chapter, so feel free to skim ahead.

The first part revolves around a fundamental question: why is there an increasing demand for setting up server-side tracking? The goal is not to ramble about how this approach is the next best thing since sliced bread, but to separate the hype from the real potential of server-side tracking for your business.

Second, we will move behind the scenes to see how server-side tracking works in contrast to the regular way, client-side (browser-side) tracking. The goal is to understand how server-side tracking can remove some limitations of tracking on the browser, and what it cannot do.

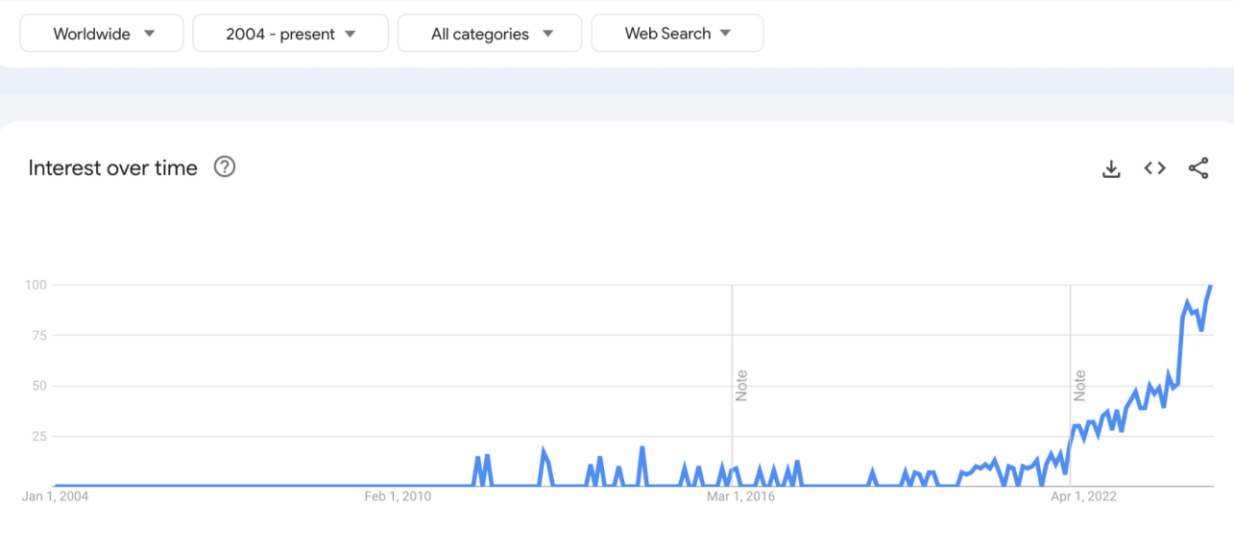
Finally, we will address some common ways to set up server-side tracking. Using Google Cloud is not the only option. Vendors with more beginner-friendly solutions are popping up every day. This section will be useful to you if you are an agency and handle multiple implementations for clients.

Part 1: Why the Sudden Interest in Server-Side Tracking?

“Data is a precious thing and will last longer than the systems themselves.”

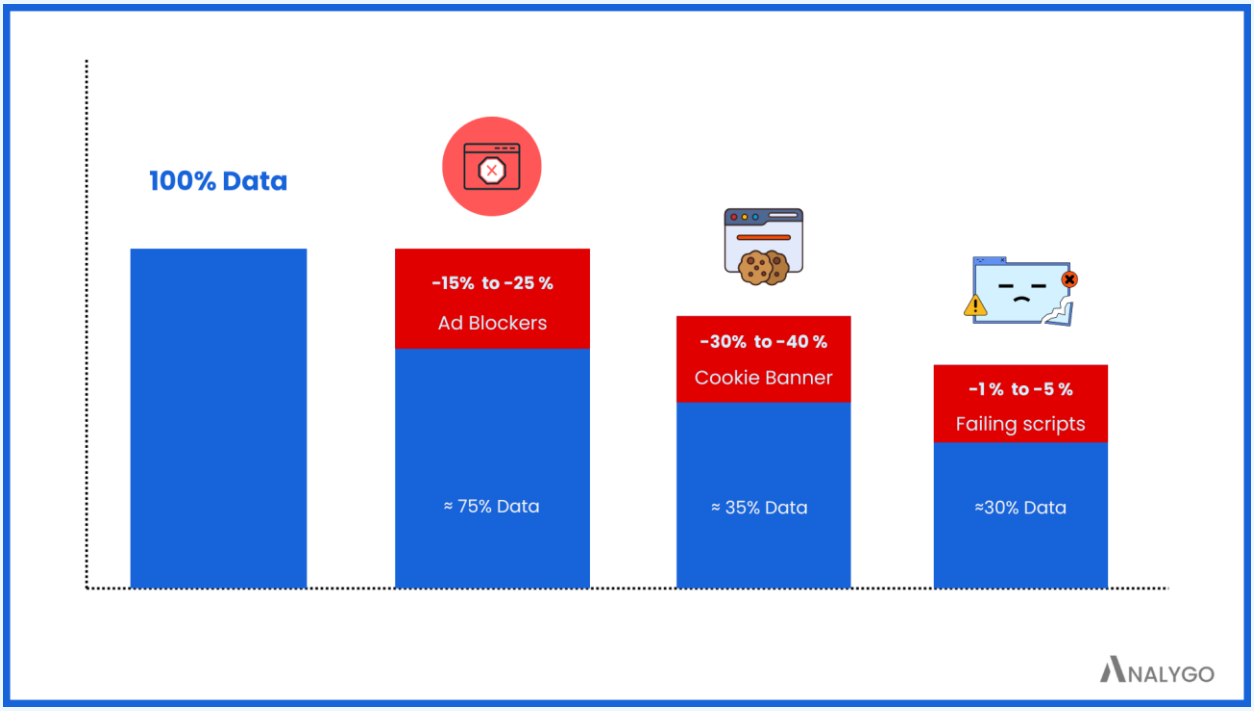
Tim Berners-Lee

Server-side tracking is nothing new. As a matter of fact, **the technology is as old as digital analytics itself**. It started to become more mainstream because of the numerous issues related to tracking on the client side.



Interest in server-side tracking has risen in the past few years

Let’s talk about some of the main factors that are fueling interest in this technology.

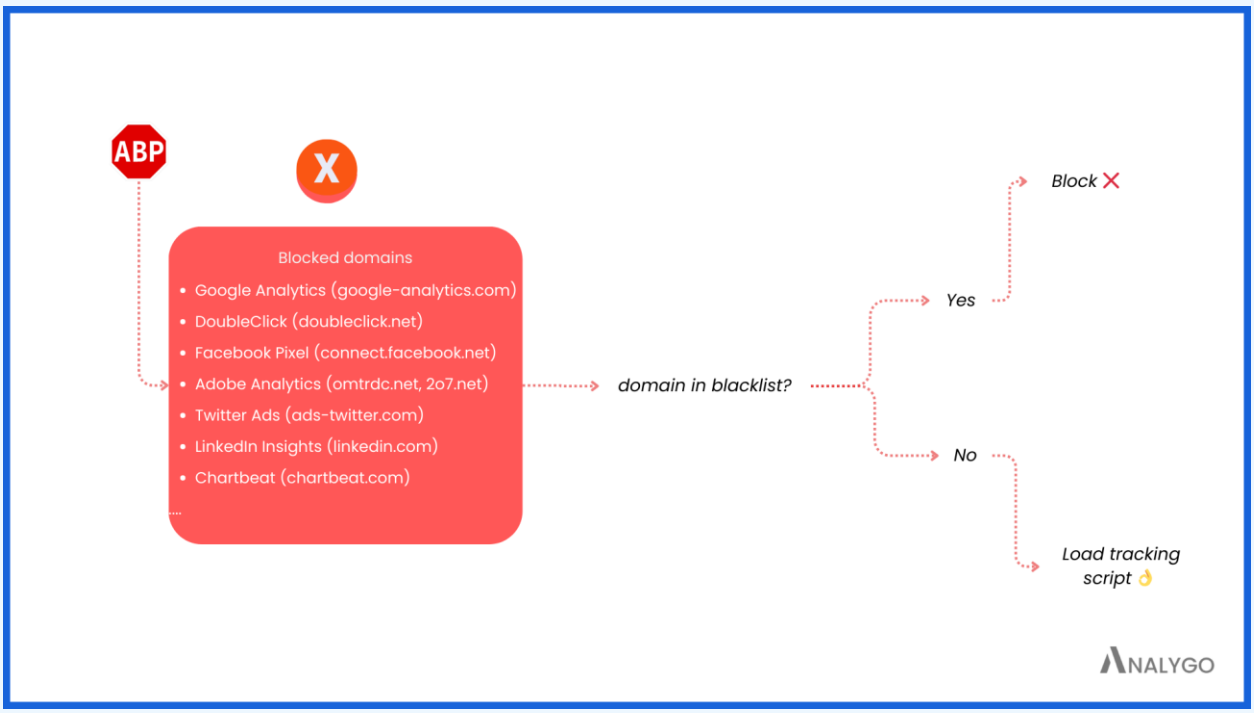


(Very) rough estimate to how much data is lost due to client side tracking

- Ad/Tracking Blockers

Around half of internet users use ad blockers, if not more. While I'm not a huge fan of intrusive popups and non-relevant ads myself, this is a BIG problem for a business trying to understand where customers are coming from.

Even if some users are okay with ads, browsers, and extensions can easily block tracking scripts.

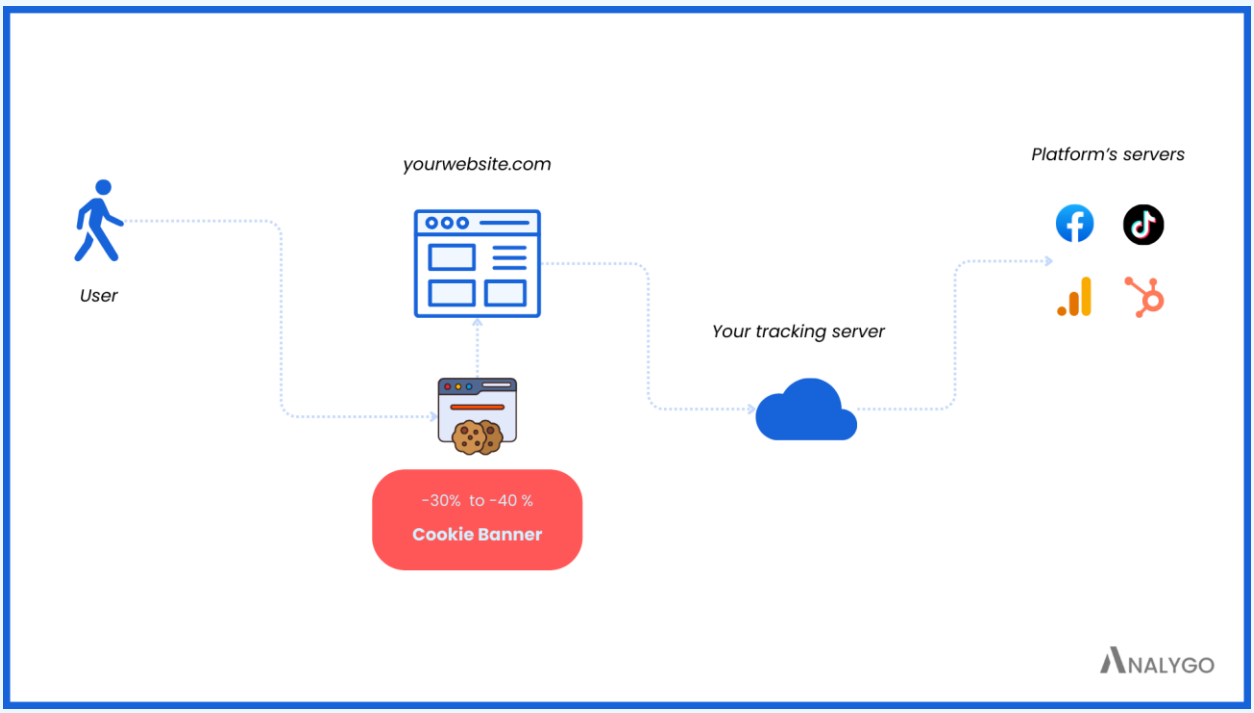


Simplified version of how ad and tracking blockers work.

- Cookie Banners and Compliance

On top of ad blockers, there is yet another barrier to getting accurate data: **cookie banners**.

While the percentage of people who agree to being tracked varies depending on the region, the average is somewhere around 30%. In regions like the EU, the percentage can be significantly higher. Saying that this number is significant would be an understatement.



Before data is sent to your server, a percentage of it is lost due to cookie banners.

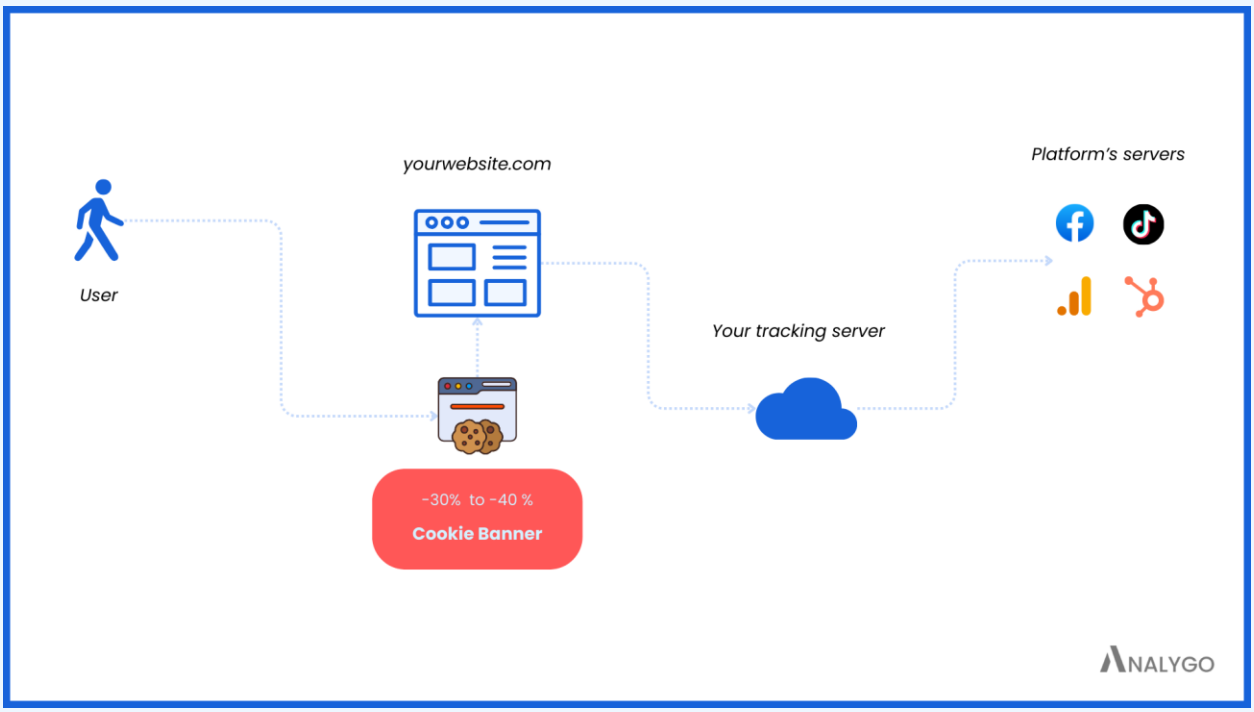
- Failing JavaScript

Sometimes your tracking code will fire too late or won't fire at all. Network issues and small changes to your website, to name but a few, can interfere with your tracking. This is something that is rarely brought up and one of the reasons tracking on the web is somewhat unreliable if the goal is 100% accuracy.

What You Can Really Do with Server-Side Tracking

After we clarified that server-side tracking is not some miracle solution, we will discuss what you can really do with it. The focus here is more on the significant advantages of server-side tracking (aka the selling points).

- Share Data with Vendors on Your Terms



With transformations, you can remove data before it reaches tags.

The issue with using so-called pixels is that you have little to no power over what the tracking script can take from your website once you inject it. Even if you don't care about the kind of data you are sharing with big tech companies, sending personally identifiable information (PII), even by mistake, can cause you a lot of headaches.

The solution is to have a buffer between the vendor and your website: **your own server.** Once the data reaches your server, you can make any necessary changes before shipping the data to the vendors. Features like transformations (more on this later) enable you to control the amount of shared data easily within Google Tag Manager.

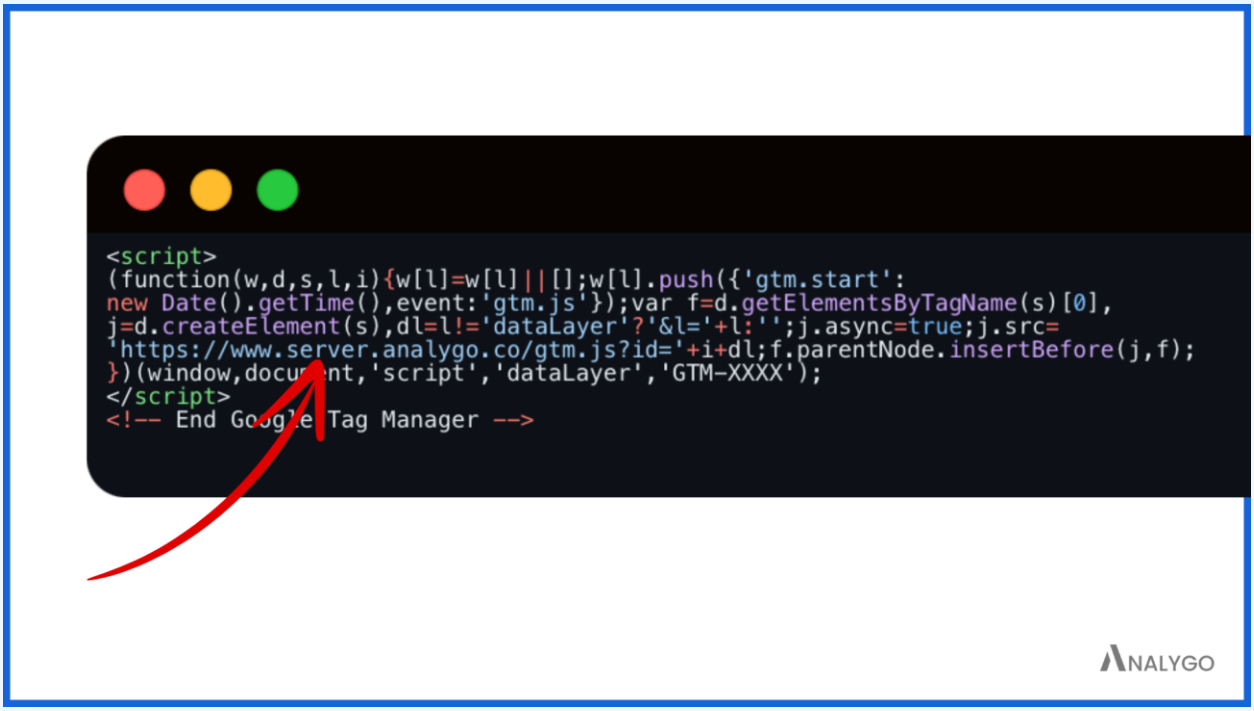
- Track Using Your First-Party Domain

The script below is something anybody who has installed GTM before is well familiar with.

The default way of setting up GTM by loading from googletagmanager.com

The part I highlighted in red is how tracking blockers can identify platforms like Google Tag Manager, GA4, etc. They will look for a pattern like ‘googletagmanager.com’ when your website’s page is loading and easily block it.

If you want to avoid this, you can use your own domain (ssgtm.yourwebsite.com, for example) to load GTM.



The alternative way of setting up GTM by loading from your own domain to bypass ad blockers.

You might ask: **“How will this solve my problem?”**

When you use your own domain to load scripts, the extensions and browsers that block your tracking code can no longer spot the tracking script as accurately.

Main Takeaways

There are other advantages that people cite when talking about a server-side setup. The two below are, in my opinion, what make the technology worth investing in:

- More control: Using your own domain to control what data you share with other vendors.

- Improved data quality: By finding a way around software that blocks ads and tracking scripts and loading GTM from your first-party domain.

As we saw earlier, getting all your data back is simply not possible with server-side tracking. However, you will see some improvement in accuracy because Ad blockers and tracking blockers can’t detect your GTM script.

Finally, think of this technology as a simple enhancement to your setup. Should you implement it? Not necessarily. It all depends on your specific circumstances and business needs (more on this later).

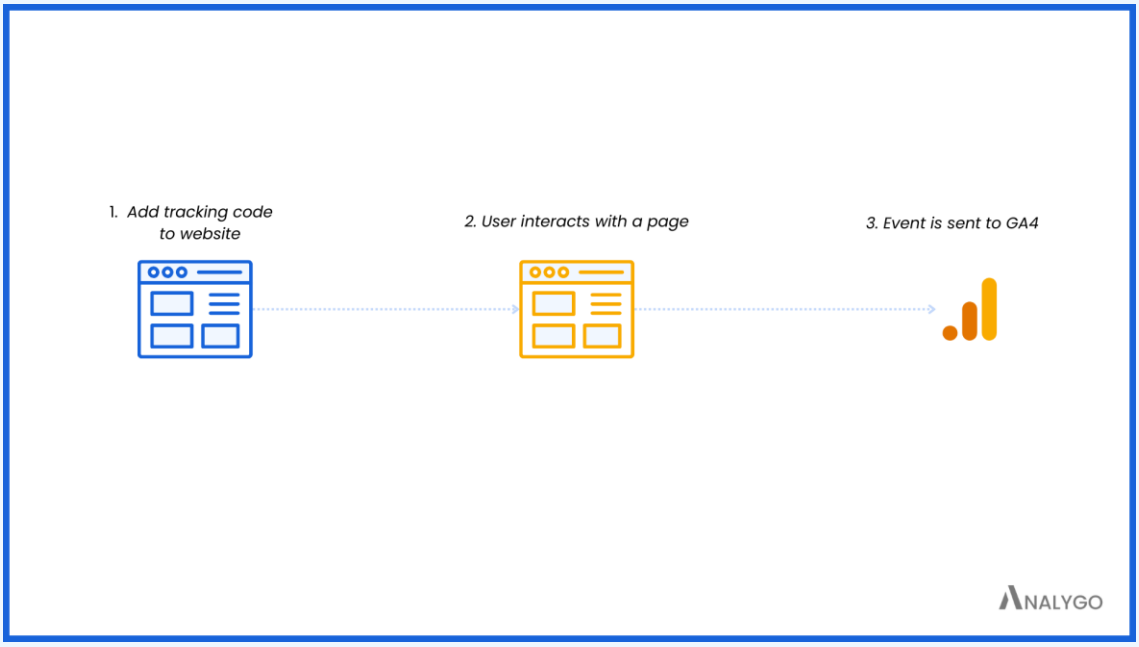
Part 2: moving from client to server-side tracking

“Data is the new oil, and the ability to store, process, and analyze data is the engine of modern business. Servers are the infrastructure that makes this possible.”

Eric Schmidt, Former CEO of Google

After explaining what you can do with server-side tracking, the next part will be about the changes in tracking approach once you move to a server environment.

How tracking on browser (aka client-side tracking) works?



Tracking on the web is straightforward: add a script, and you are good to go.

The nice thing about a client-side approach to tracking is it's quite straightforward. Let's take the example of Google Analytics.

1. Add JavaScript code

It doesn't matter whether you are using Google Analytics code (gtag.js) or working with Google Tag Manager, a code links Google Analytics with your website and listens to interactions.

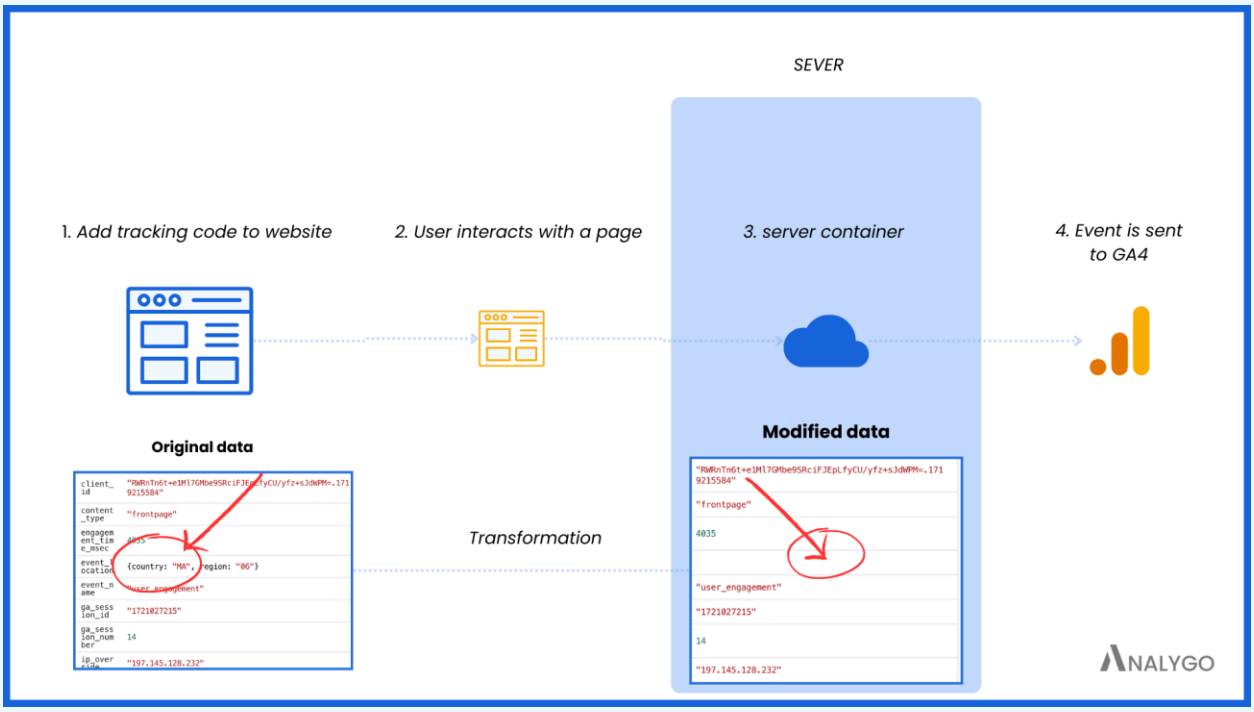
2. Data collection

The script will then gather all kinds of data from the browser language to the number of products bought.

3. Process and display data

Each interaction is sent as an event to Google's servers. It will be processed, then displayed in the UI reports.

How server-side tracking works?

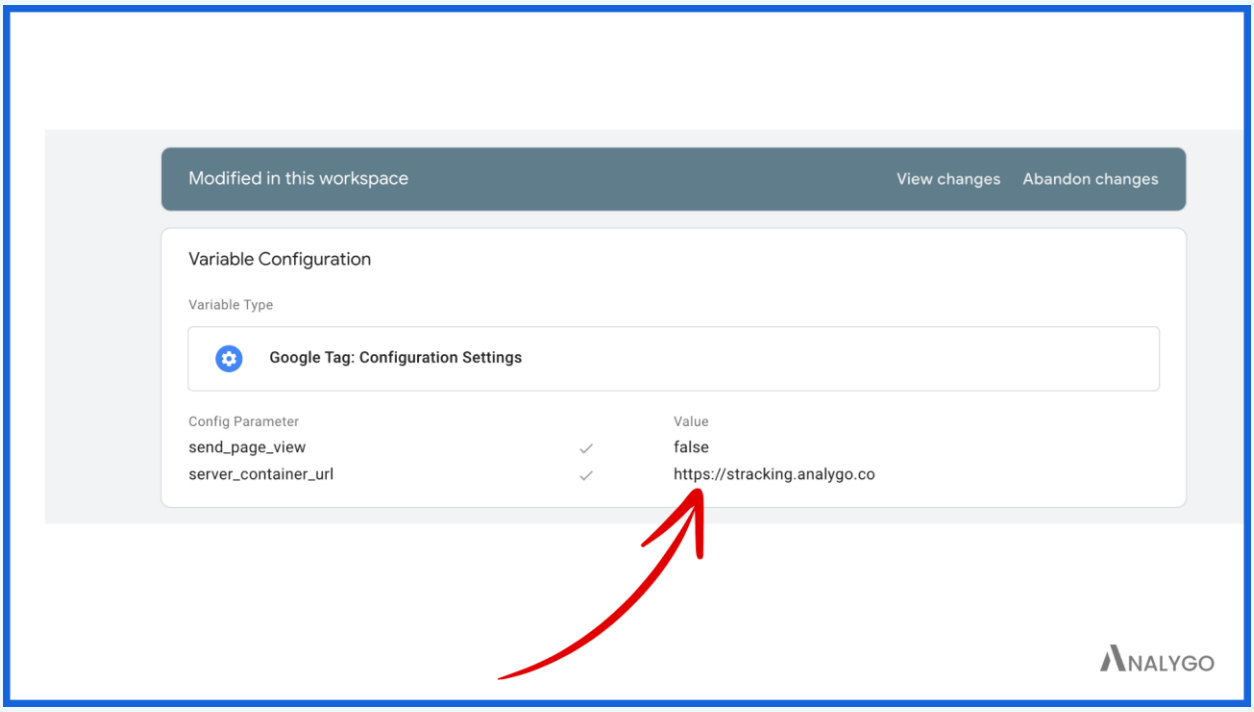


Simplified version of the server-side tracking process.

The keyword here is ‘server’. Using the previous example of Google Analytics, once the data is collected, instead of sending it directly to Google’s servers, it will go through our own server for extra processing (removing sensitive data for example).

1. Add your server’s URL to tracking script

Here we simply redirect the data collected to our server. This can be done using Google Tag Manager by adding the URL as a parameter (’server_container_url’). Alternatively, you can modify the gtag script you are not using GTM.



Example of server URL added to a client Google tag configuration settings

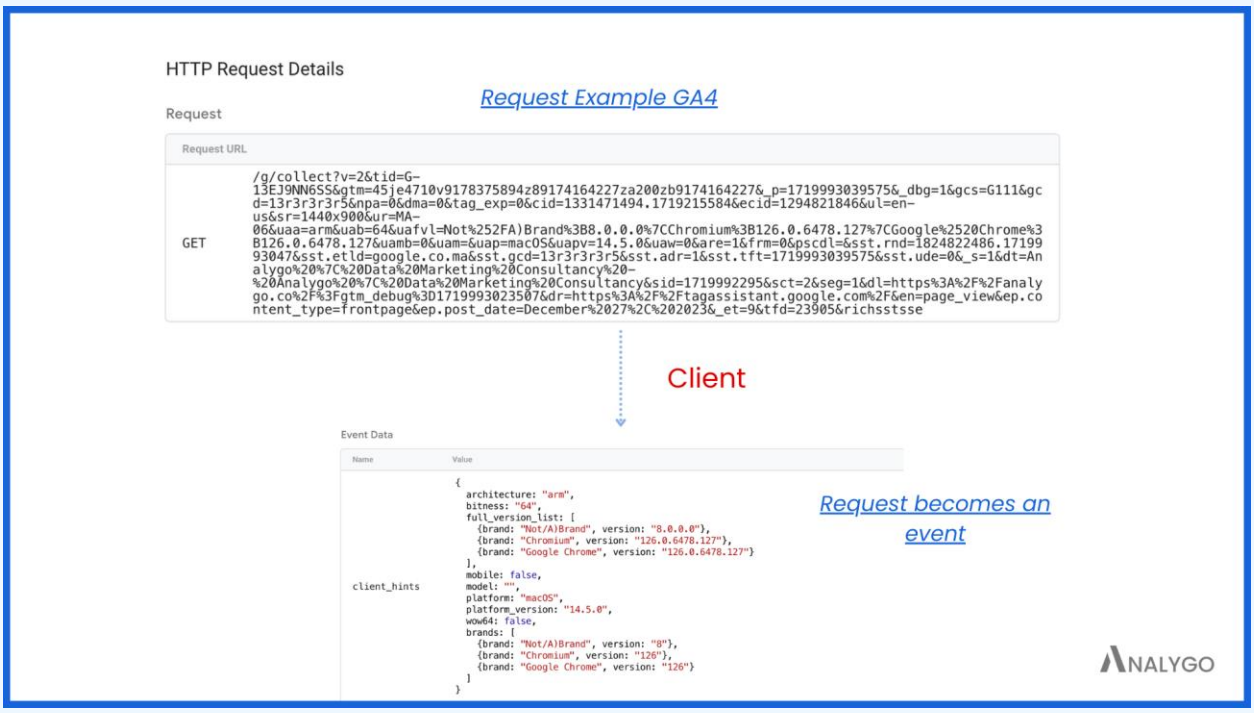
2. A request is sent to server

You can think of a request as a **message** you send to the server. The message contains the data collected in the browser.

3. Clients claim request

The client is a concept specific to server-side tracking. I like to think of it as a **translator or decoder**. It will take the request and turn it into an event. Here we are in a familiar

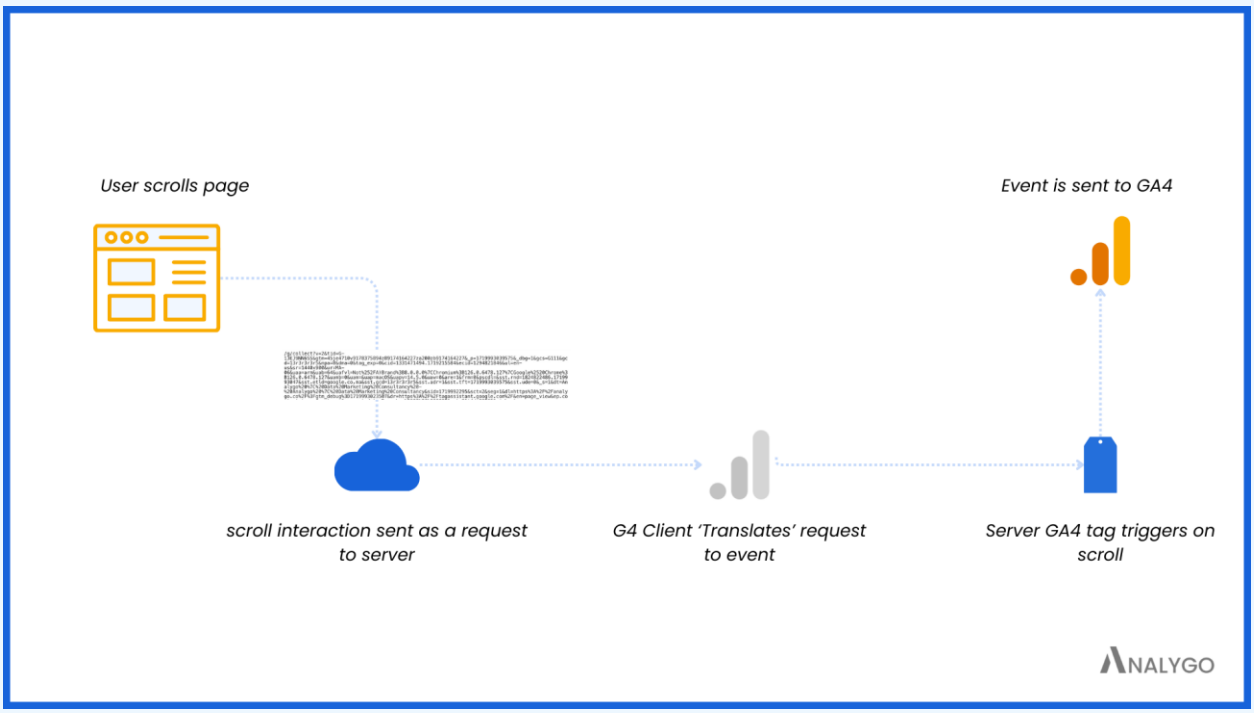
territory. Events like a page view can be used as triggers for our tags.



4. Server tags transfer data to vendor's server

Once events are available to our server container, we can use triggers, variables to set up our tags. The only thing that will stand out to you at this step is **transformations**.

Events from the browser come with a lot of baggage. Information about the user's browser, location will be sent along with the tag unless you decide not to. That's where transformations can be quite handy.



Summary of the server-side tracking process.

Main Takeaway

There are more technical nuances that were omitted in this chapter for the sake of providing a picture of how server-side tracking is different from the browser approach. If you retain one word, it should be "**buffer**".

It's all about introducing an extra filter between your website and your vendor. This gives you leverage over the kind of data you share.

Part 3: Setting up server-side tracking

“Efficiency is doing things right; effectiveness is doing the right things. There are countless ways to be effective, and the best method depends on the specific circumstances and objectives.”

Peter Drucker, management consultant

There are many ways to set up server-side tracking, some rely on Google cloud to host the tracking server, others delegate the complex task of maintaining the server to dedicated vendors like Stape, Addingwell, taggers, etc.

There is of course a right way to go about it. Which one? **The one that fit your circumstances**. Ultimately the decision will boil down to a combination of costs, expertise, and convenience.

Cloud run

Cloud run is the default way to set up server-side tracking, if you choose to host your tracking server with Google. Previously, we had a different technology called app engine, which was less ideal for smaller websites due to costs (although it can be great in some cases).

Cloud run is comparatively cheaper depending on your set-up. But money is only a part of the issue: **you still need someone who understand how to keep things running smoothly,** which leads us to the second option.

Other hosting options

You can deploy server-side tracking in whatever environment that fits your needs.

For instance, I work with a lot of agencies that don't necessarily have the time, nor the expertise to maintain a server-side tracking implementation for their clients over the long term. I usually recommend using one of the hosting solutions below, unless their client is a large business with an internal team to maintain the set-up.

- Stape

Stape is definitely one of the top options for deploying server-side tracking. If you are an agency with the goal of streamlining your server-side implementation for clients, then you should absolutely check them out.

- TAGGRS

If you want to quickly set up server-side tracking, TAGGRS is undoubtedly one of my top recommendations. They have a ton of templates available for download to your container, which is what makes them really standout from similar solutions.

- Addingwell

Addingwell are great as well, and more popular in the EU as the go-to solution for deploying serve-side tracking. I've done a few implementations with Addingwell, and what I like about them in particular is their great support team

Which hosting option to use?

There are no strict rules on when to choose either option, but consider the following If you are a:

- Large business seeking maximum control and flexibility
- Frequent user of Google Cloud Platform
- Team or agency available to maintain your tracking

In this case, deploy your tracking server using **Cloud Run**. It is the most versatile solution we discussed.

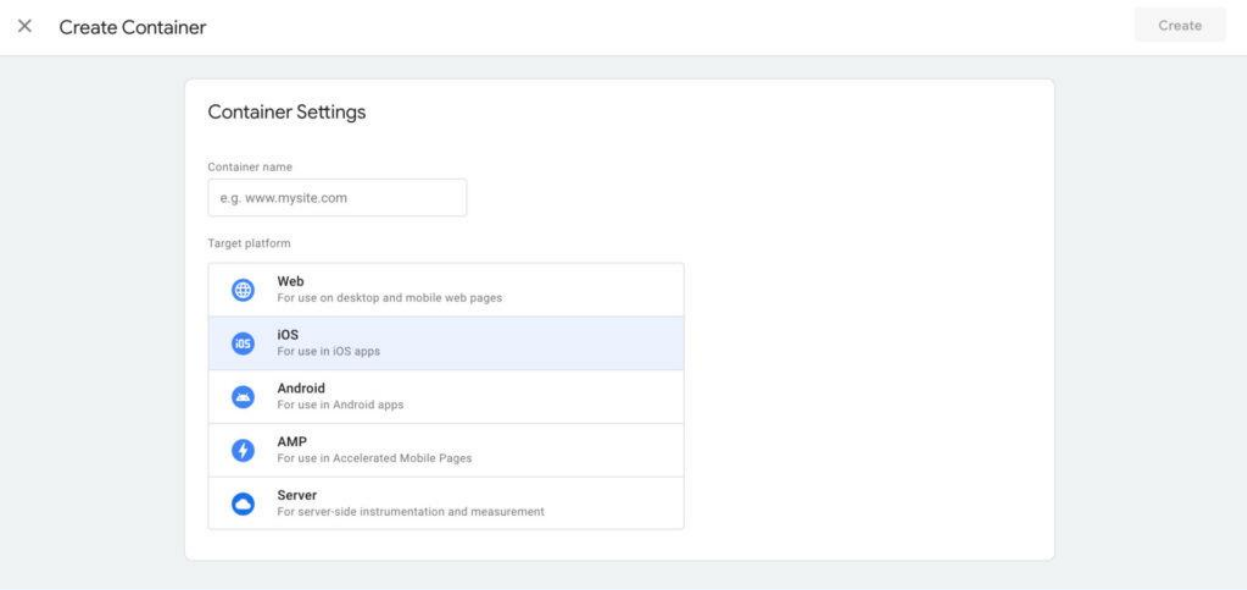
On the other hand, if your primary concern is cost, and you don't want to spend a lot of time managing your setup, a solution like Stape is probably more suited for you.

In the next section, we will explore some ways you can implement server-side tracking. I will start with the standard way using Cloud run before moving to other hosting options like Addingwell and Stape.

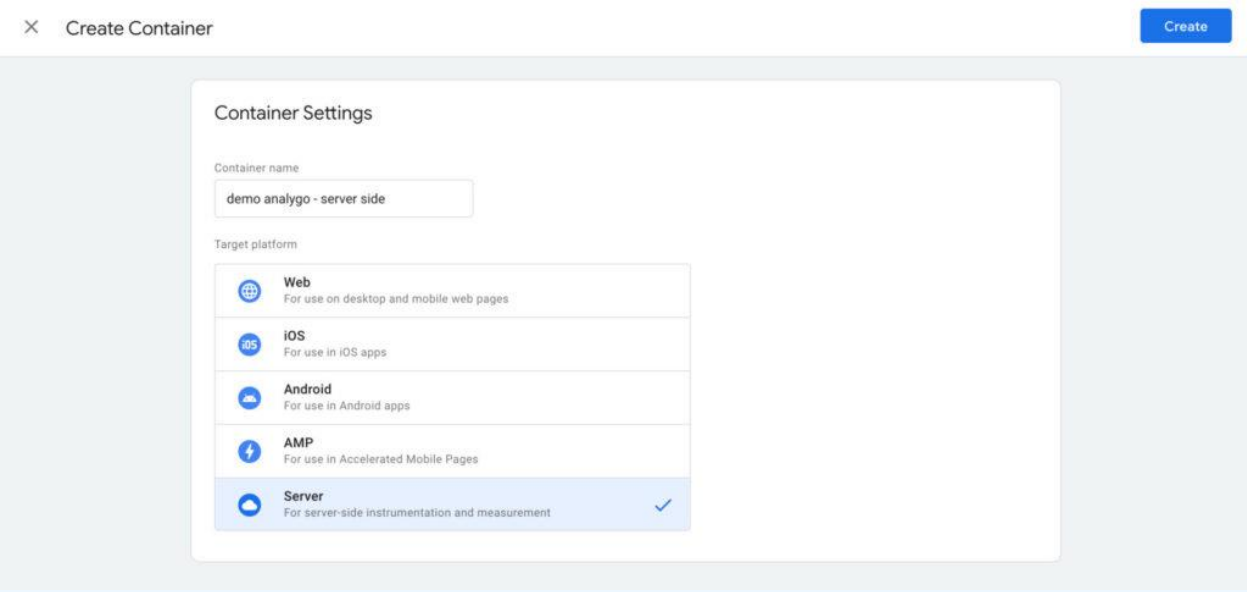
Server-side tracking with Cloud run

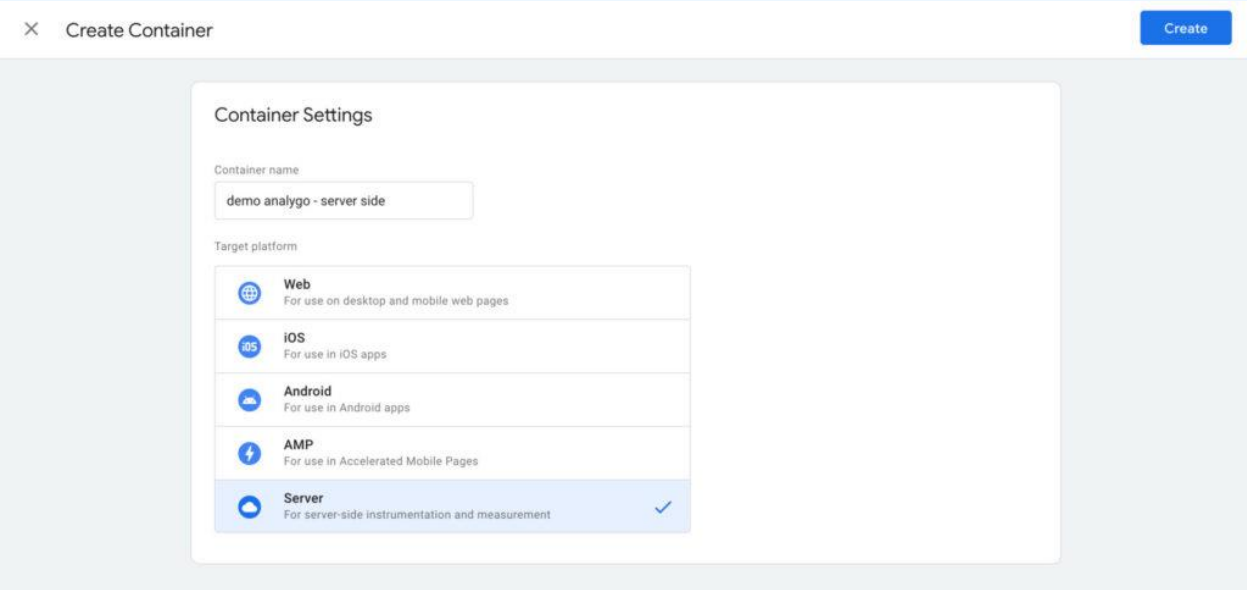
Step 1: create server-Side container

There are many types of containers in GTM. Up until now, you probably used only the web container.



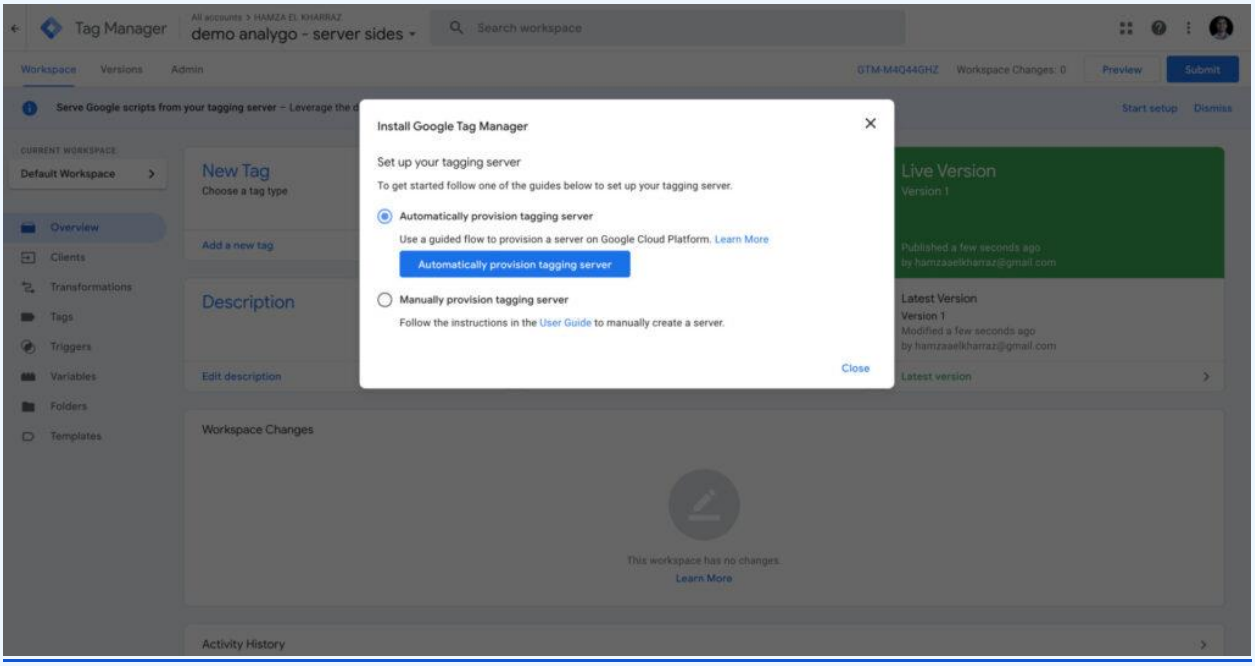
It’s time to create a server container. Go to your GTM account > Admin > Create Container > Server.



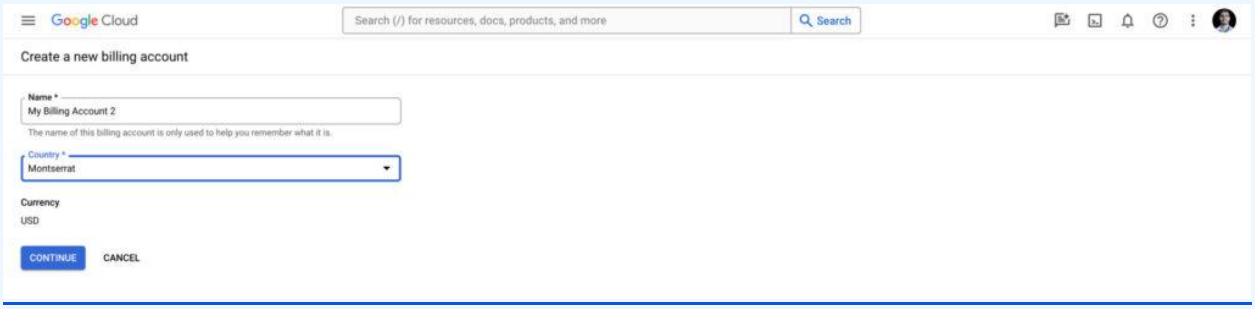


Step 2: provision the Tagging Server

The tagging server will forward the data you collect to platforms like Facebook. Before, you sent this data directly using a tracking pixel.



Select the Automatic provisioning option. Make sure to add your billing account as well.

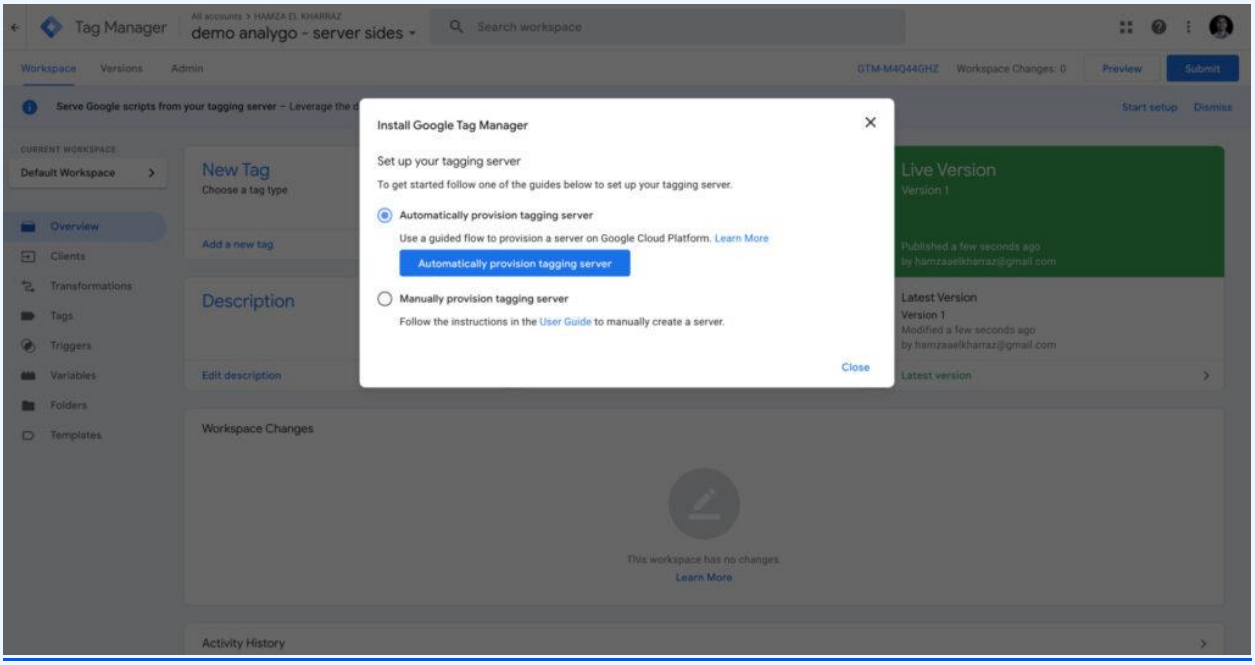


Note: This configuration is free until you increase the number of servers to handle production level traffic.

Step 3: Test Your Deployment

You will see a window with your server's information. The URL of your tagging server will look like this:

https://server-side-tagging-[XXXXXX]-uc.a.run.app



Test the tagging server by adding “/healthy” to your tagging server URL. You will see an “OK” if everything is working as expected.

https://server-side-tagging-[XXXXXX]-uc.a.run.app/healthy

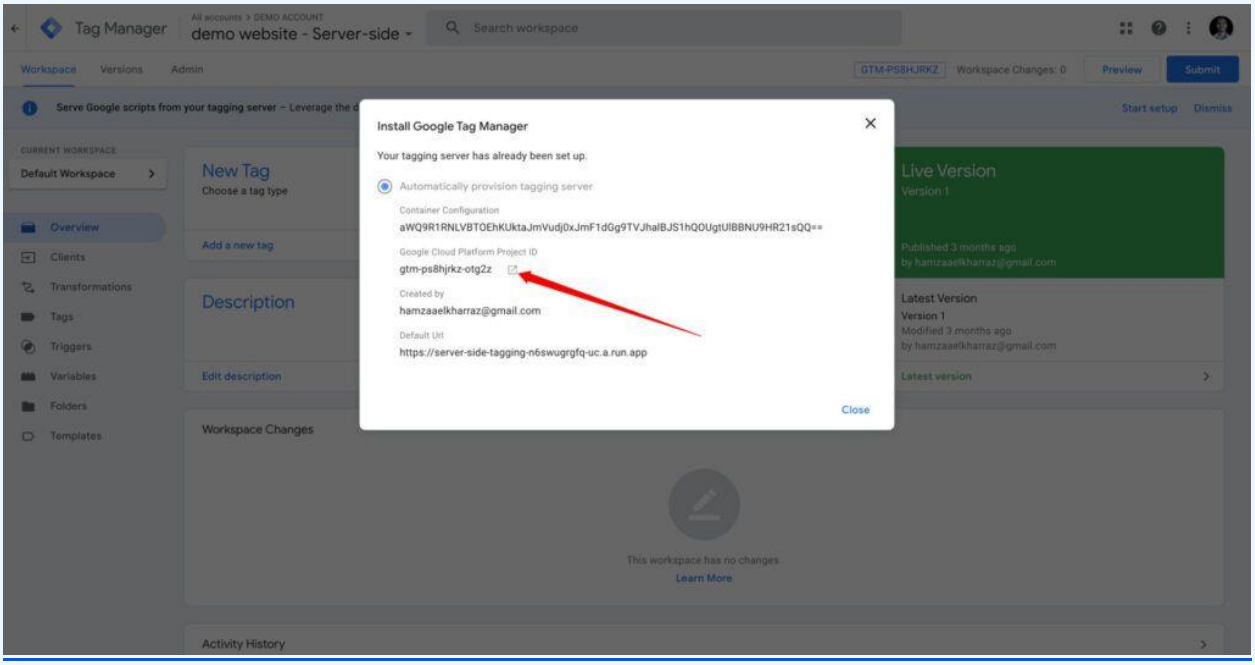
Note:

- Google will set up a server for you in the ‘us-central1’ region. You can change this setting later on.
- This configuration is for testing purposes only. You will not be charged until you increase the number of servers (more on this later on).

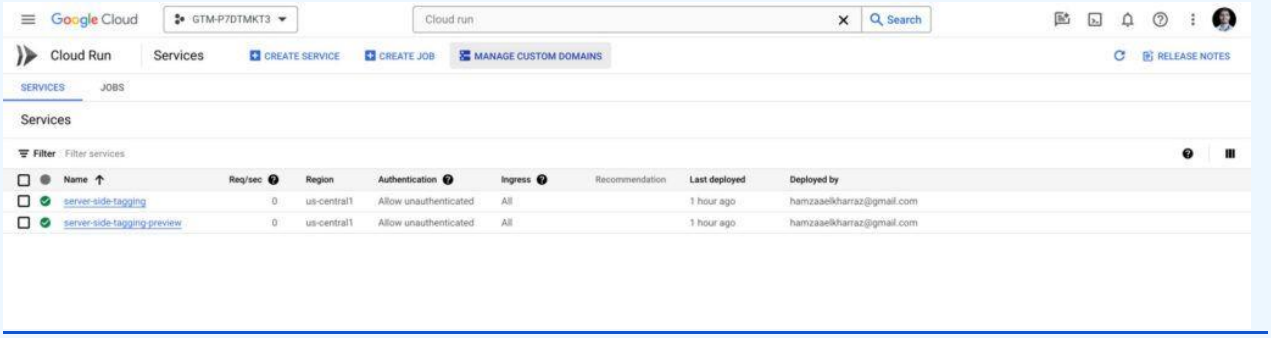
Step 4: Set Up A Custom Domain

The next step is to replace the default URL with our own custom domain. The data you will collect after this process is considered first-party. As long as you keep using the URL provided by Google, the tracking will be done in a third-party context.

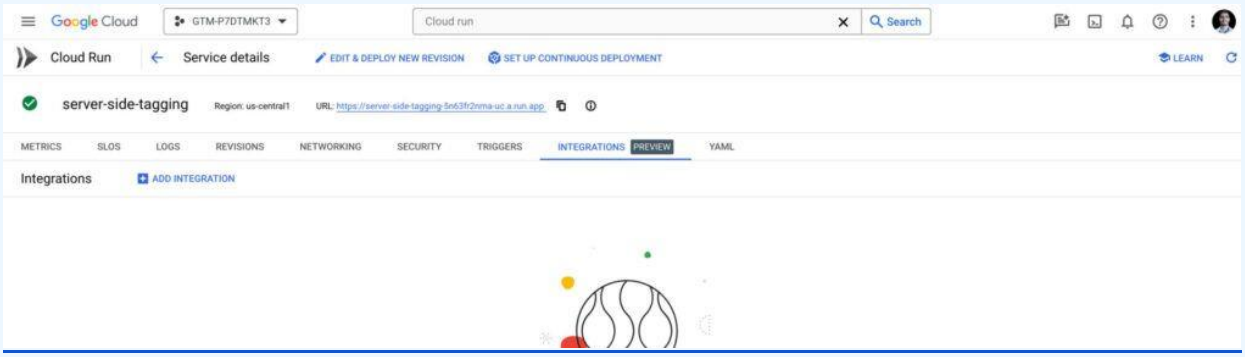
Go to your Google cloud project by clicking on the icon next to your **Project ID**.



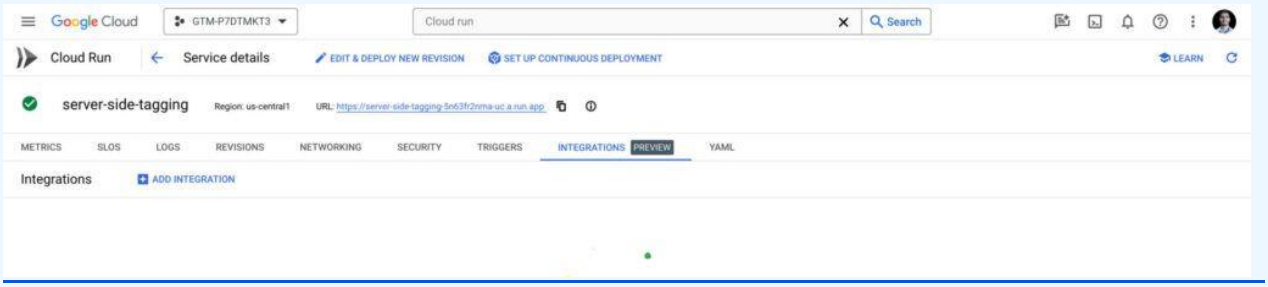
In the search bar, look for “cloud run”. You will land on a page containing your tagging and preview server.



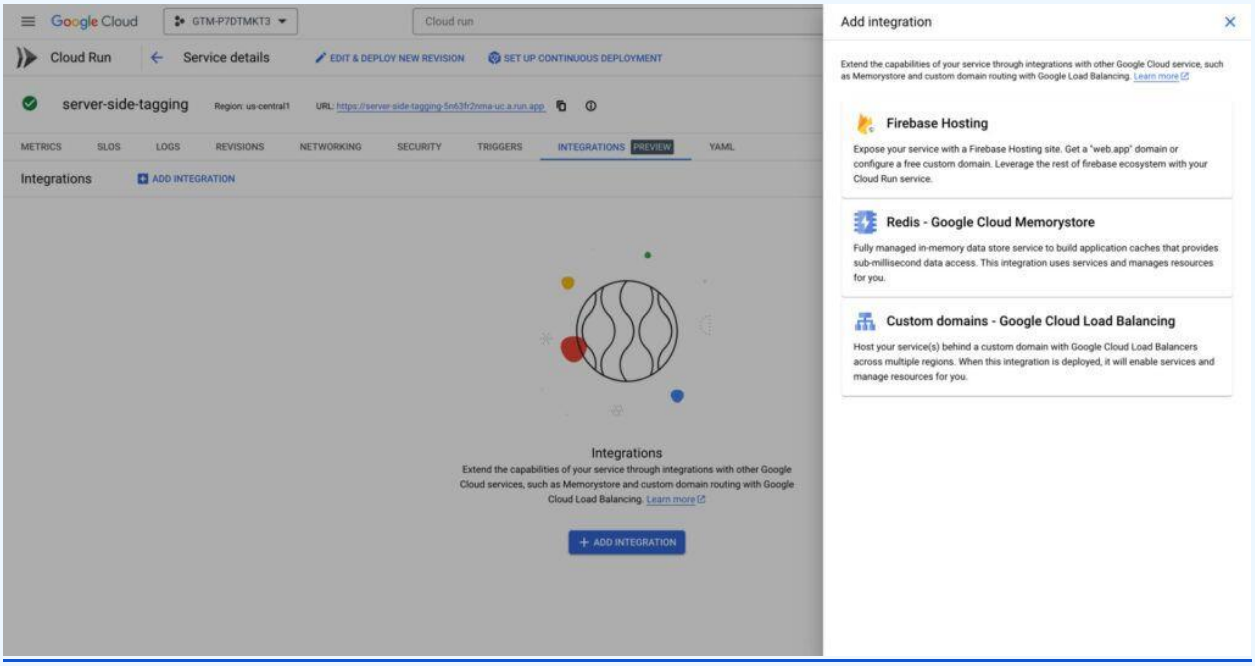
Select “server-side-tagging” and browse to integrations.



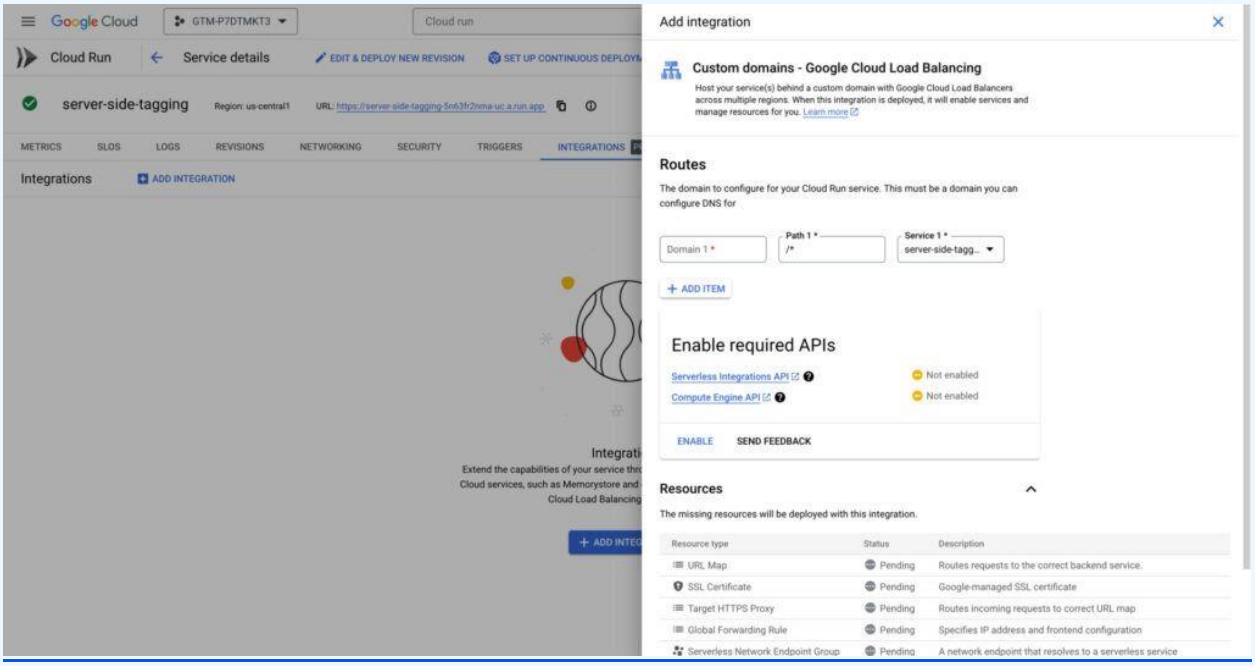
Click on add integration.



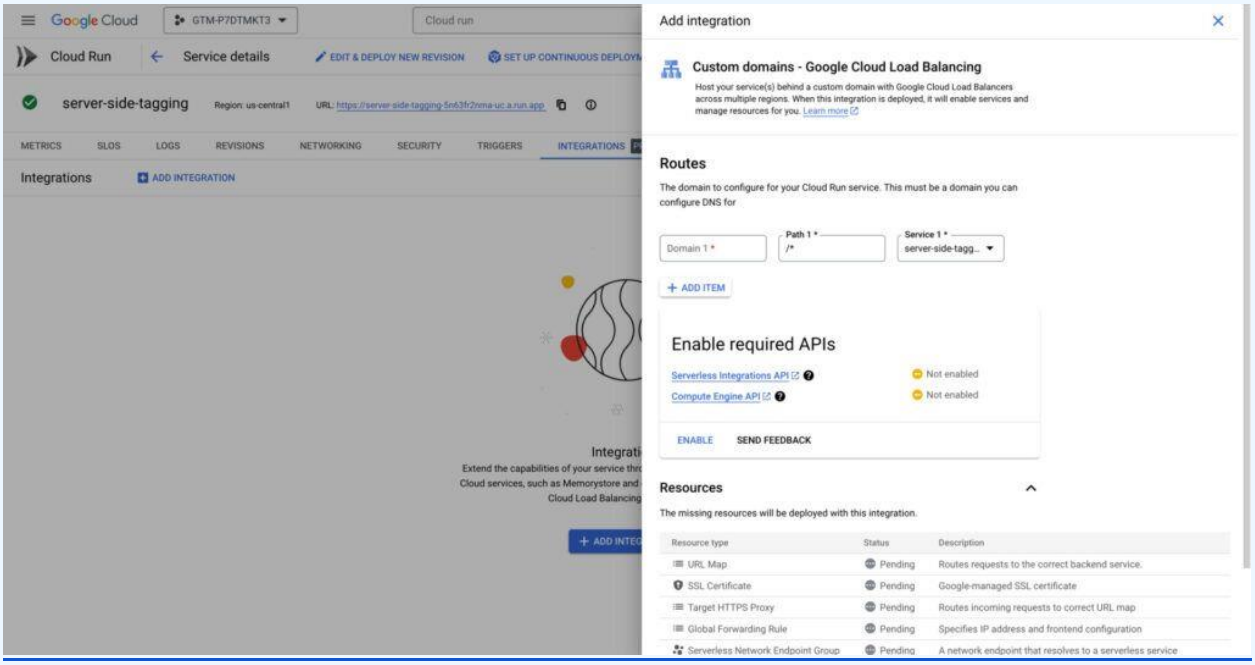
Select **Custom domains – Google Cloud Load Balancing**



Add a new domain in our case we use ssgtm.analygo.co. make sure you are using a new subdomain just for the purposes of server side tagging (you can use this format: **ssgmt.yourdomain**)

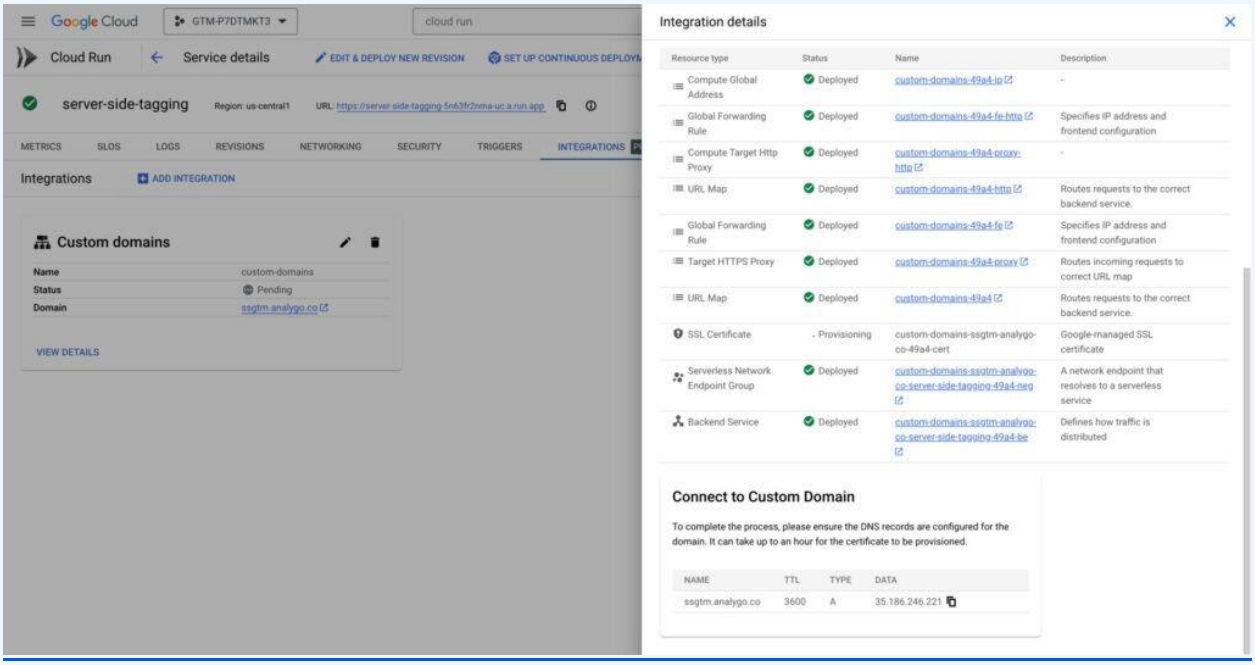


Enable the two APIs.



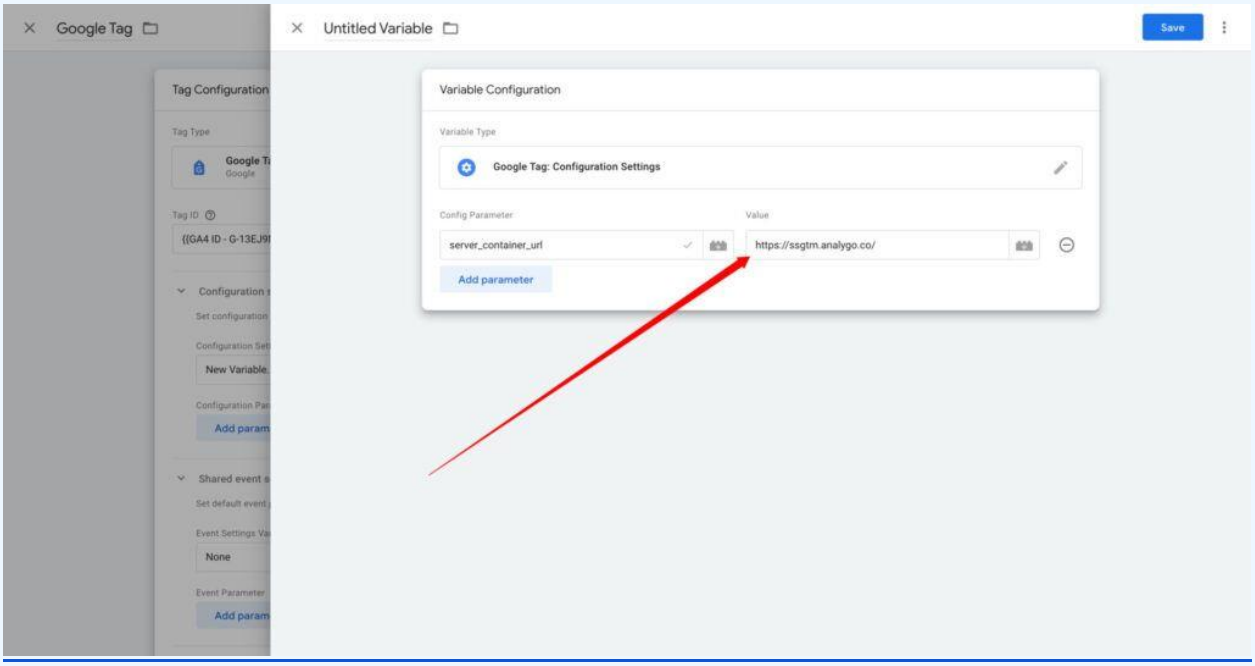
You need to go to your hosting provider and add the A records.

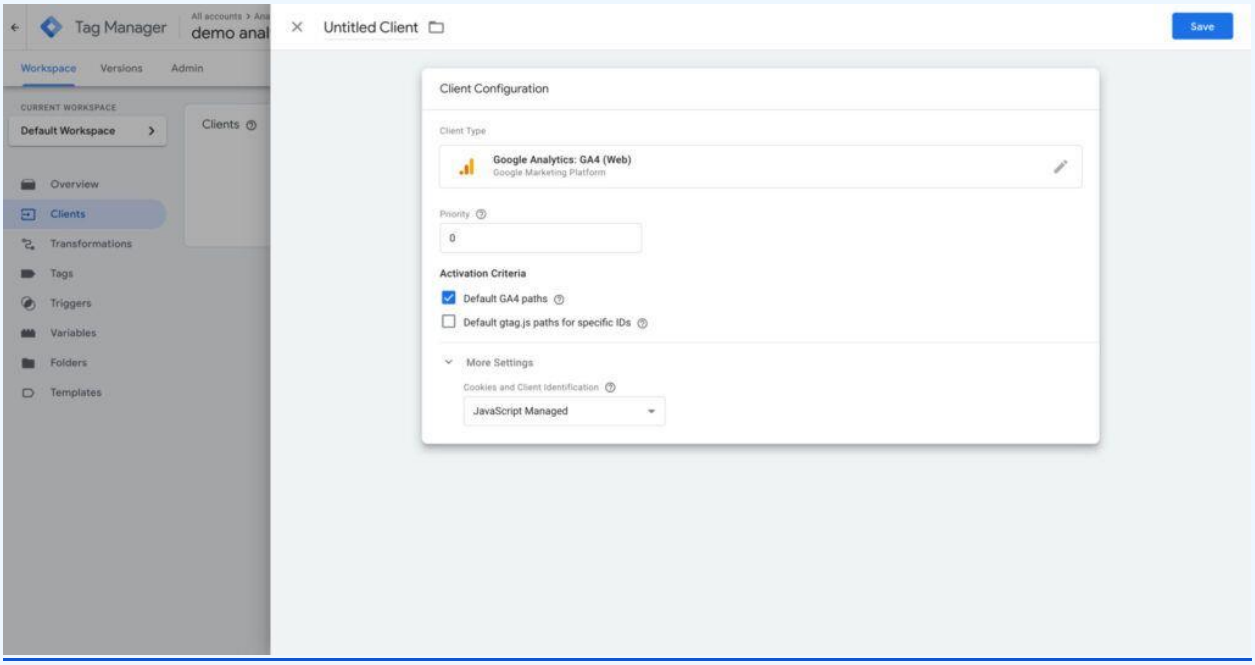
Note: the SSL certificate will stay on pending until your domain is verified. This might take up to 24 hours or more.



Finally, browse to your **server container > Administration > Container settings**.

Replace the old URL with the new custom domain (i.e., https://ssgtm.analygo.co/)

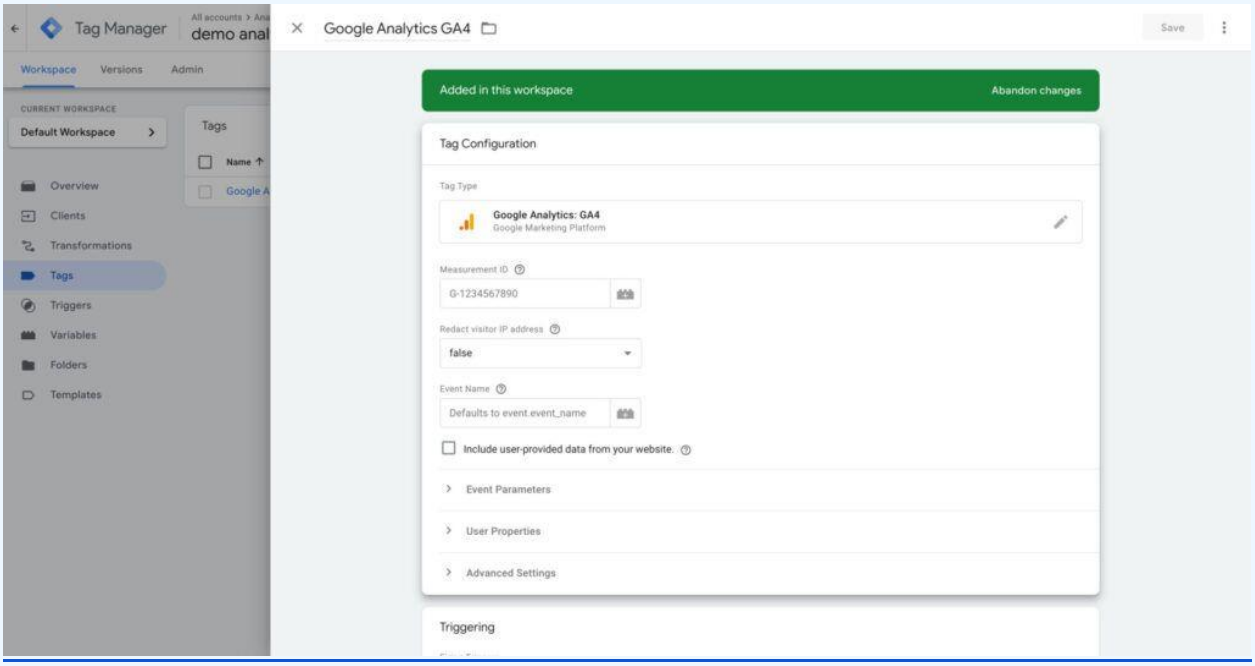




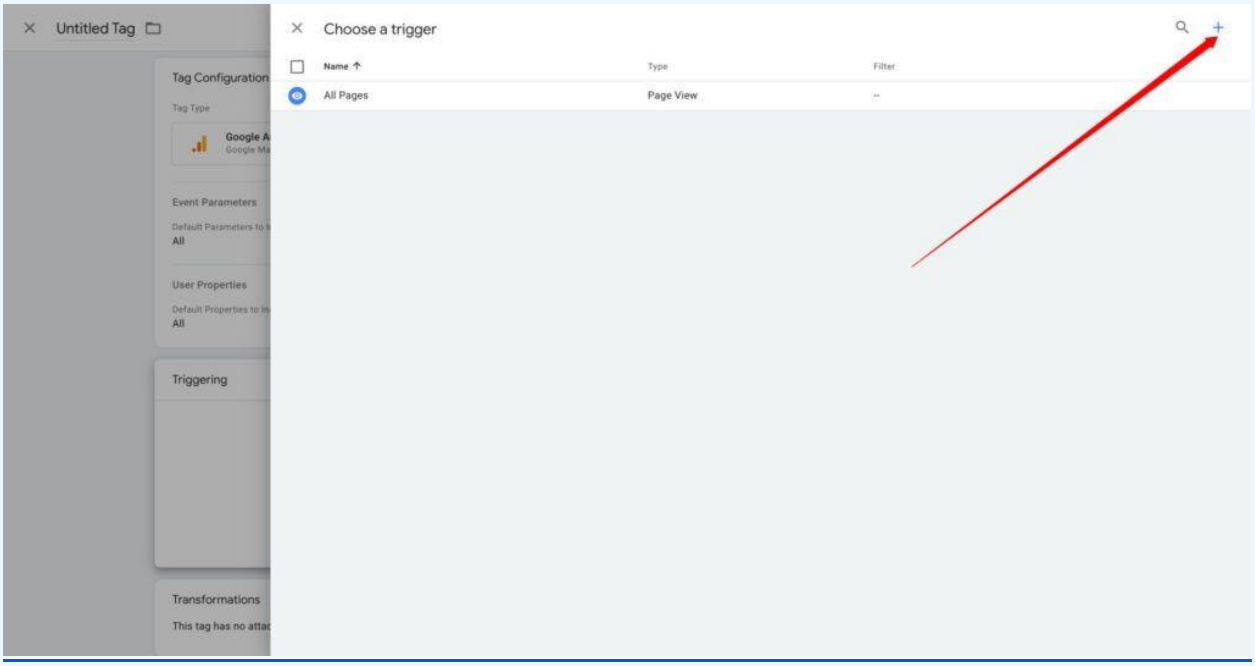
Step 6: Adding Your First Server-Side Tag

Tags receive the processed data from the client and fire according to the trigger. They work the same way web tags do.

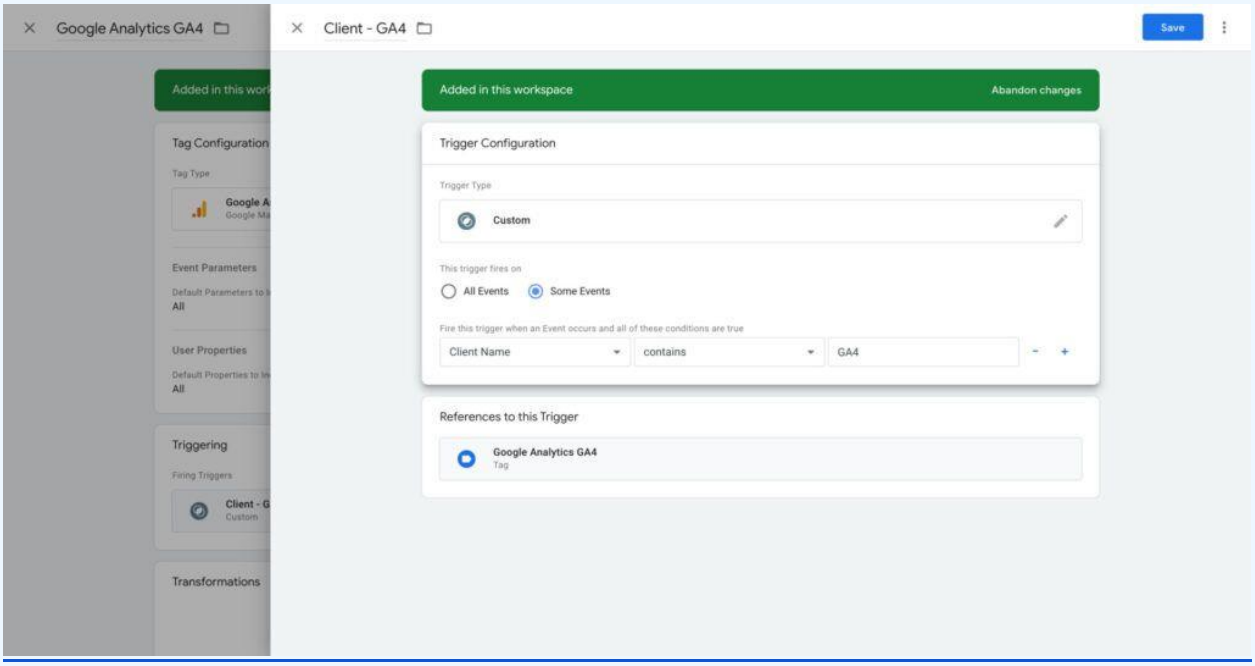
Add a new GA4 tag and leave all fields with default values.



Add a new trigger and click on ‘+’.



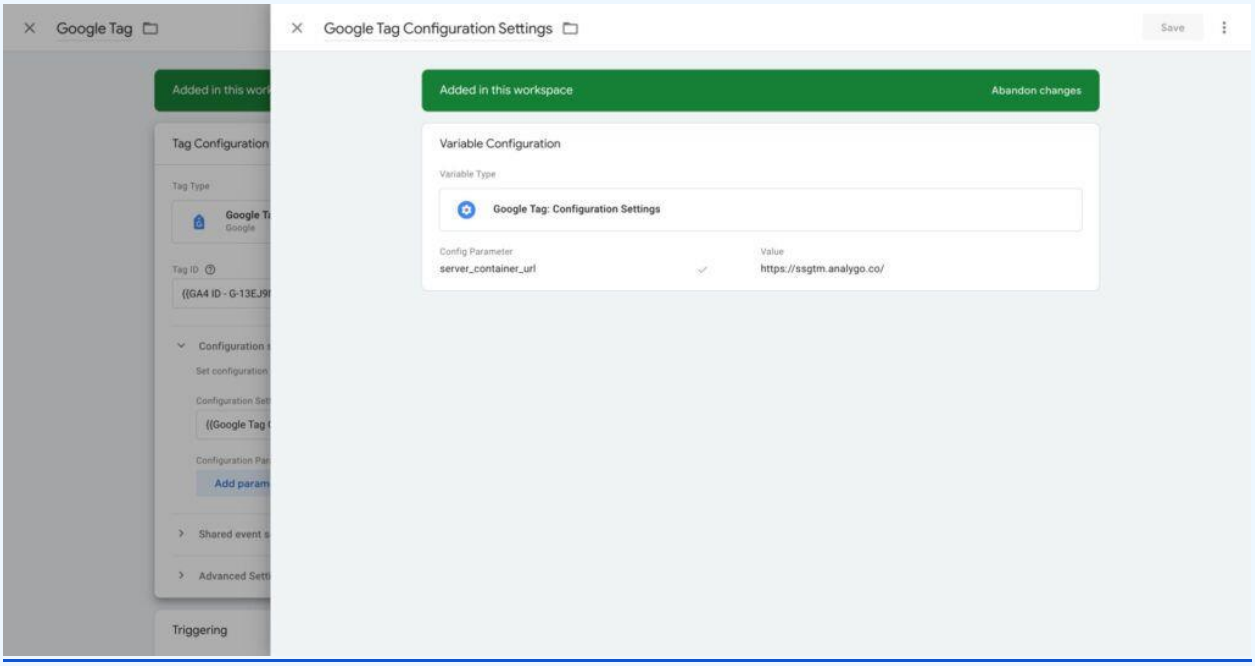
Select Some events, and Choose Built-In Variable from the drop-down. Add the Client name variable.



Step 7: Configure Your Web Tag Container

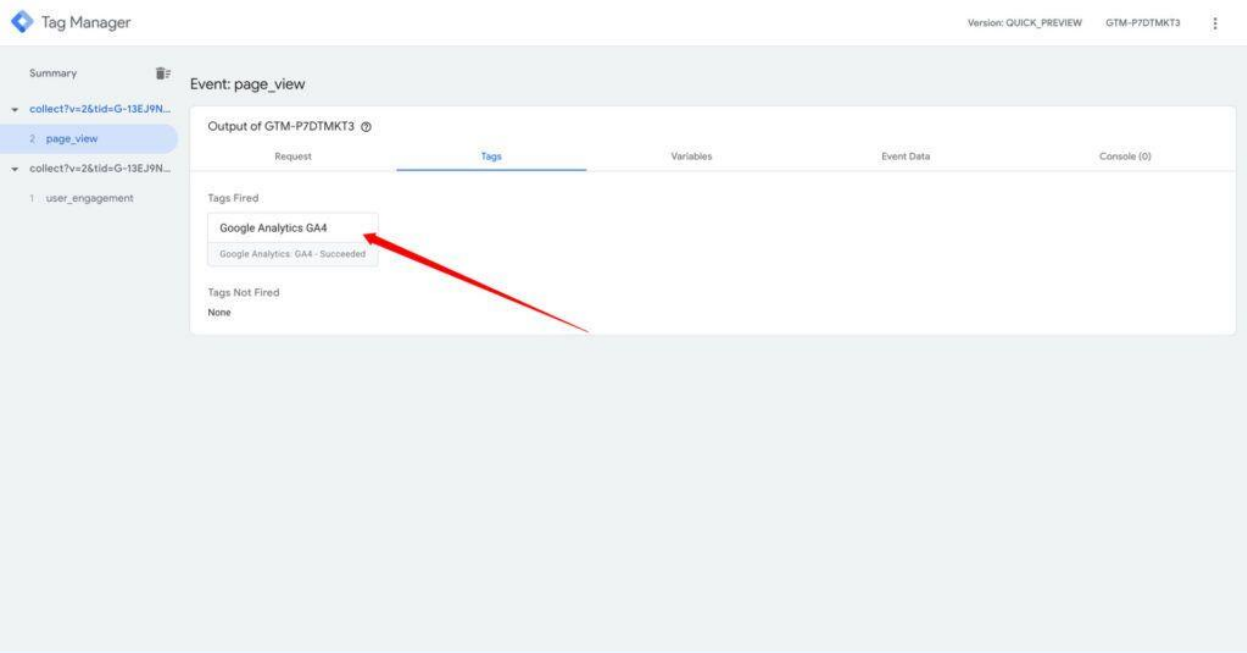
We need to add the URL of our server to the GA4 configuration tag. Open your GA4 configuration tag and select New variable under Configuration Settings Variable.

Note: we are now going to switch to the web container, so make sure you are not using server GTM



Step 8: Test Your Deployment

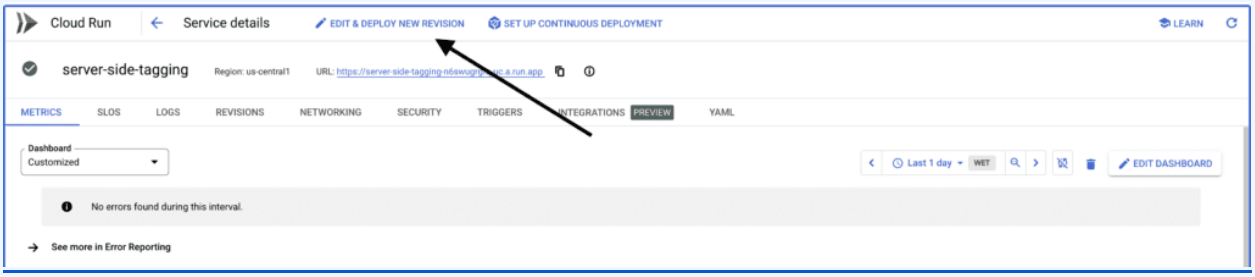
Now it’s time to test if everything is working as expected. Open preview mode in **both server and client containers**. As you can see, the GA4 tag is working as expected.



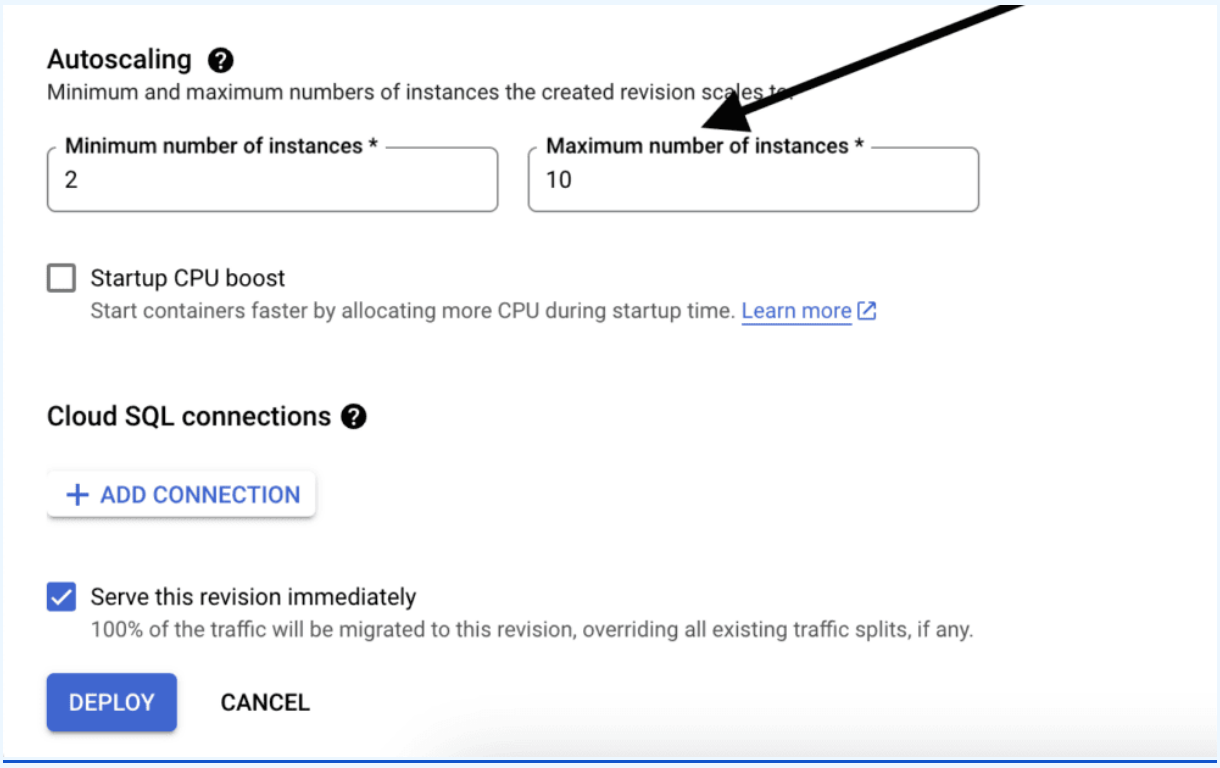
Step 9: Increase The Number Of Servers

The last thing we need to do is to make our set ready to hand handle more traffic.

Go to the integrations section in Cloud run and select **server-side-tagging**. You need to click on **EDIT & DEPLOY NEW REVISION**.



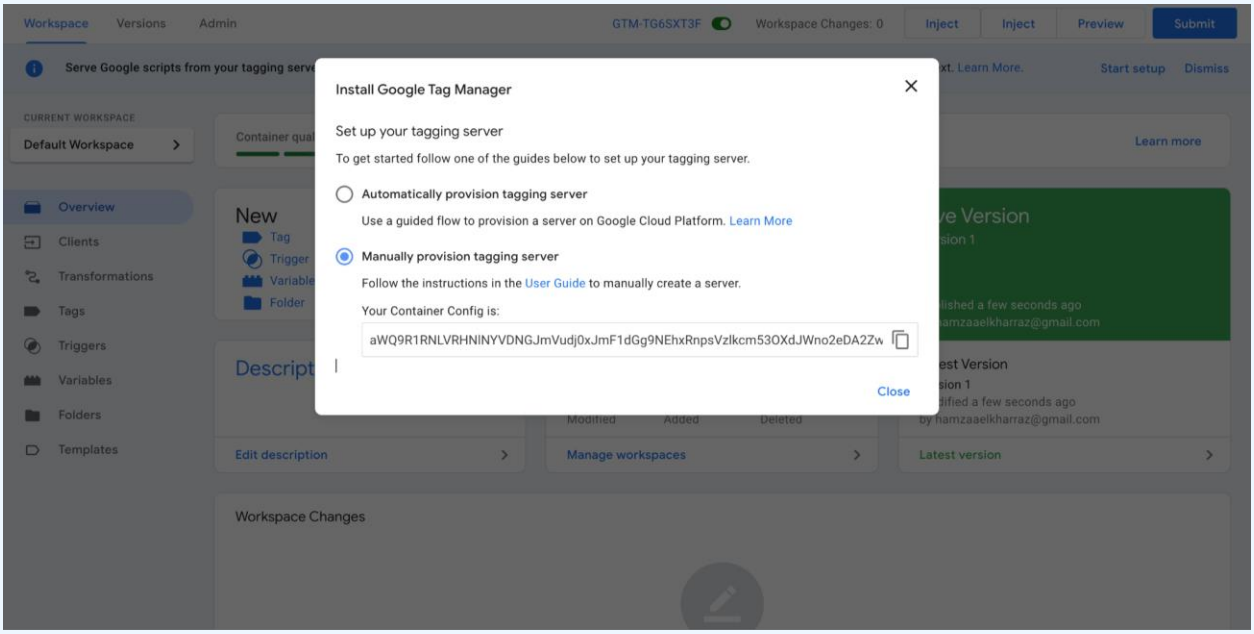
Scroll all the way down to Revision autoscaling. Change the minimum to 2 and up to 10 instances. Finally, you can click on Deploy.


Autoscaling
Minimum and maximum numbers of instances the created revision scales to
Minimum number of instances *
2
Maximum number of instances *
10
Startup CPU boost
Start containers faster by allocating more CPU during startup time. Learn more
Cloud SQL connections
ADD CONNECTION
Serve this revision immediately
100% of the traffic will be migrated to this revision, overriding all existing traffic splits, if any.
DEPLOY
CANCEL

Server-side set up with TAGGRS

Step 1: manually provision server-container

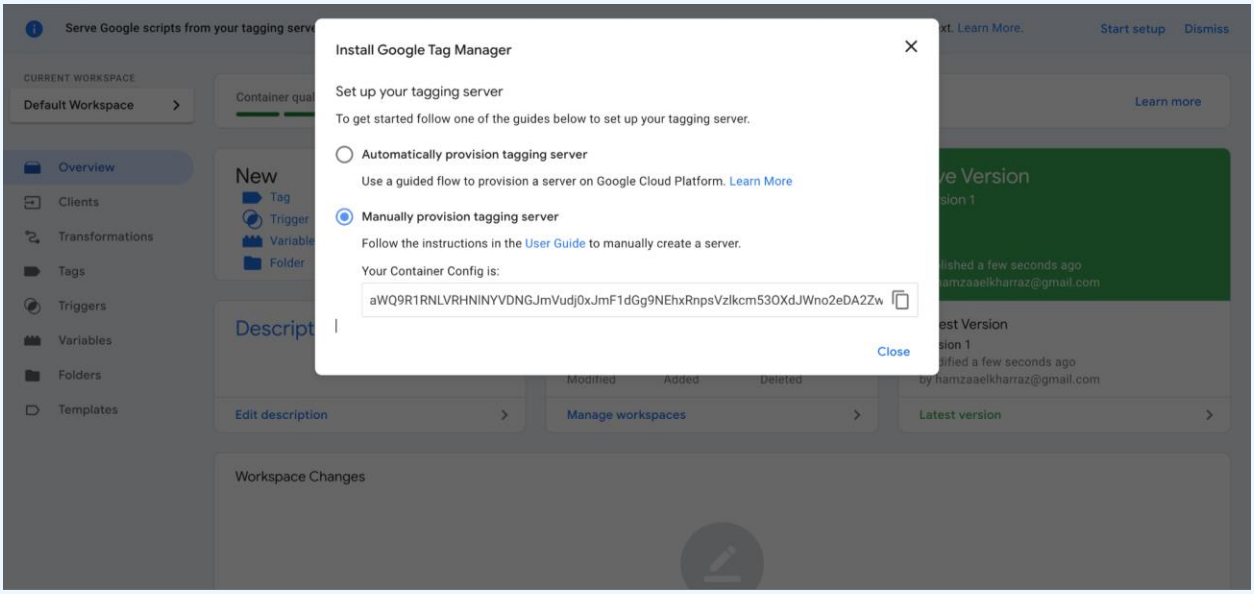
Create a new server container. Select ‘Manually provision tagging server’. Copy the link.



Manually provision the server and keep this tab open.

Step 1: manually provision server-container

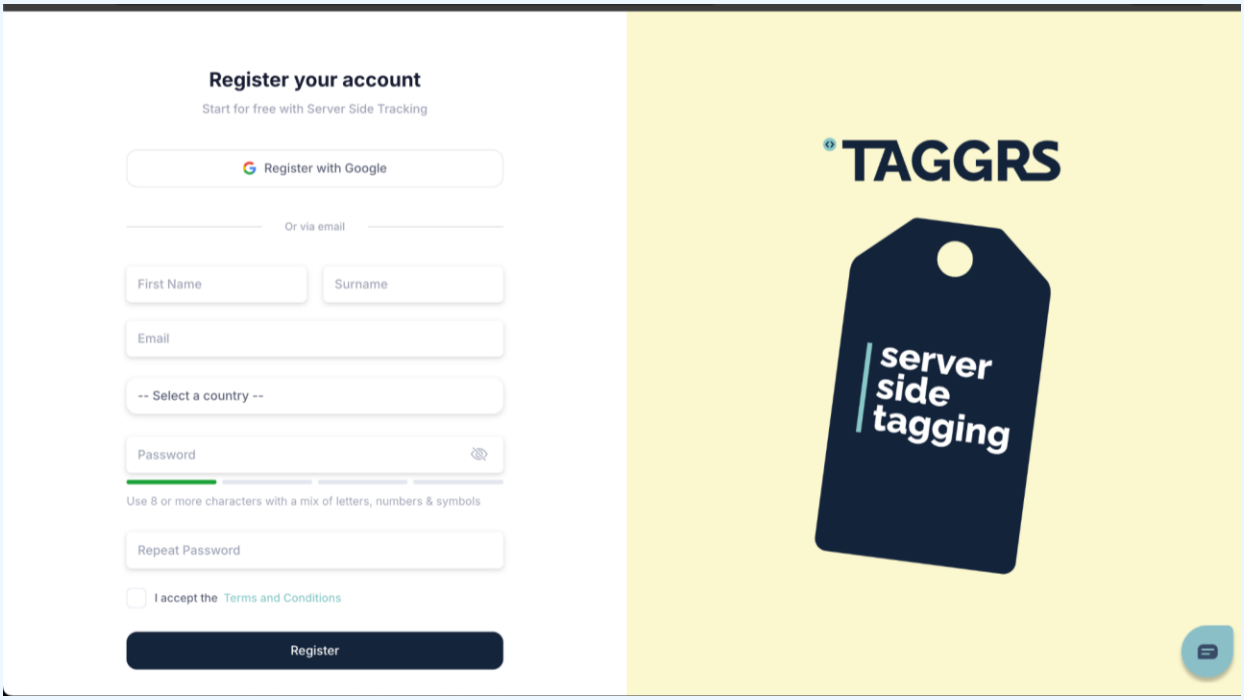
Create a new server container. Select ‘Manually provision tagging server’. Copy the link.



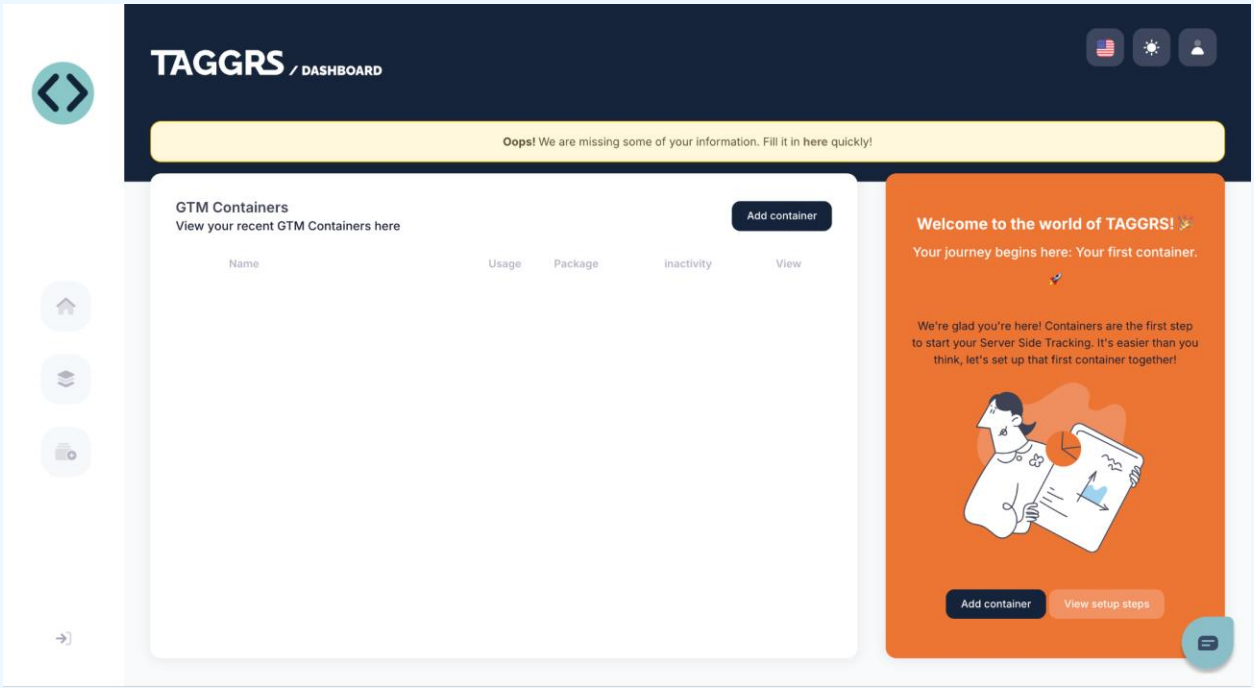
Manually provision the server and keep this tab open.

Step 2: create a TAGGRS account

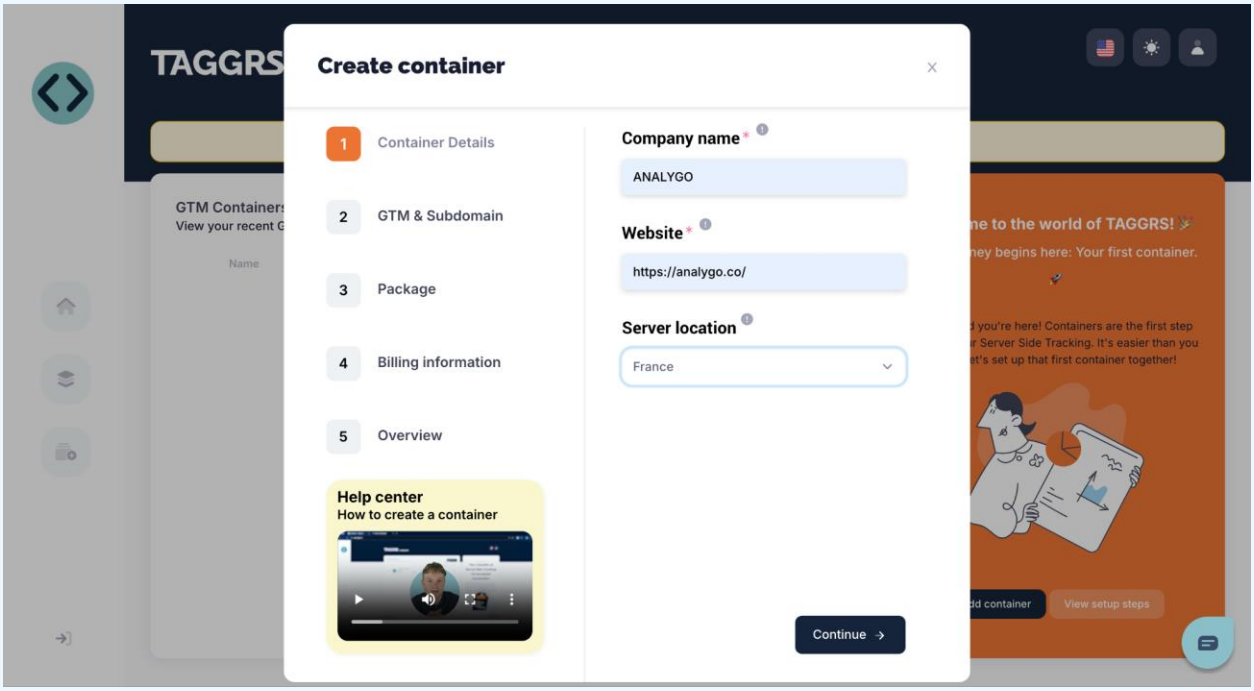
Browse to TAGGRS sign-up page and create a new account.



After signing up, you will land on this dashboard.

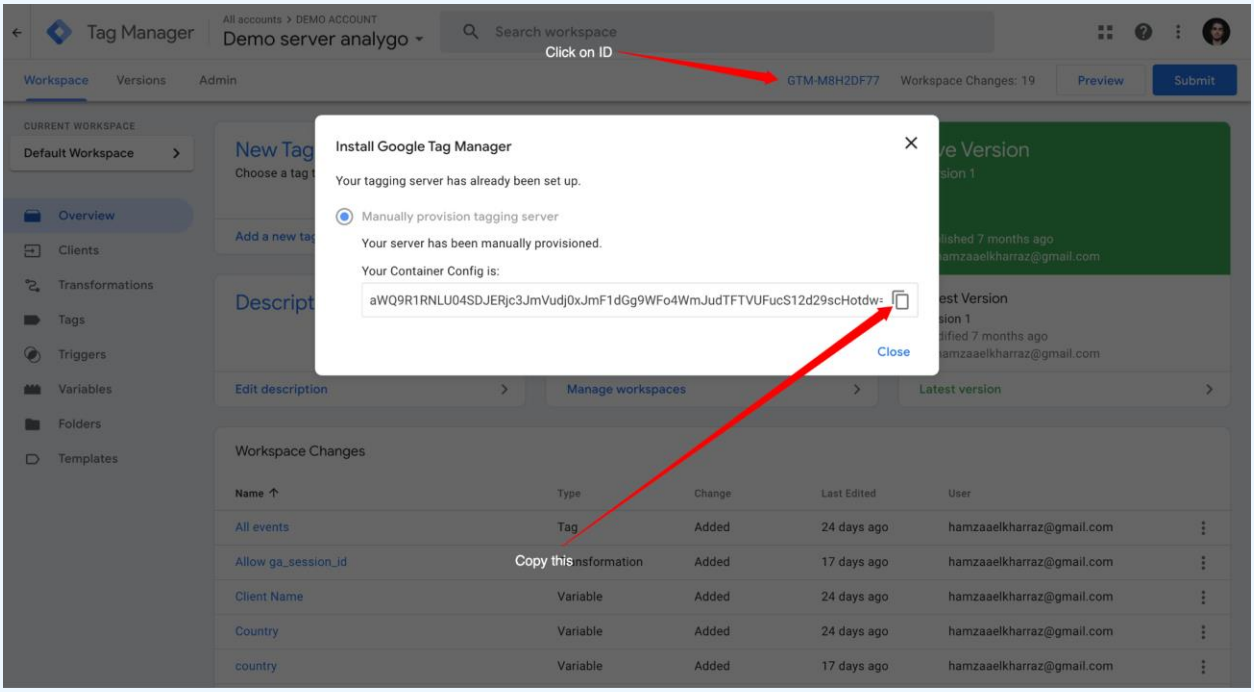


Now, let’s create a container with your GTM server container information. Click on Add container and start filling the form with your container’s details.

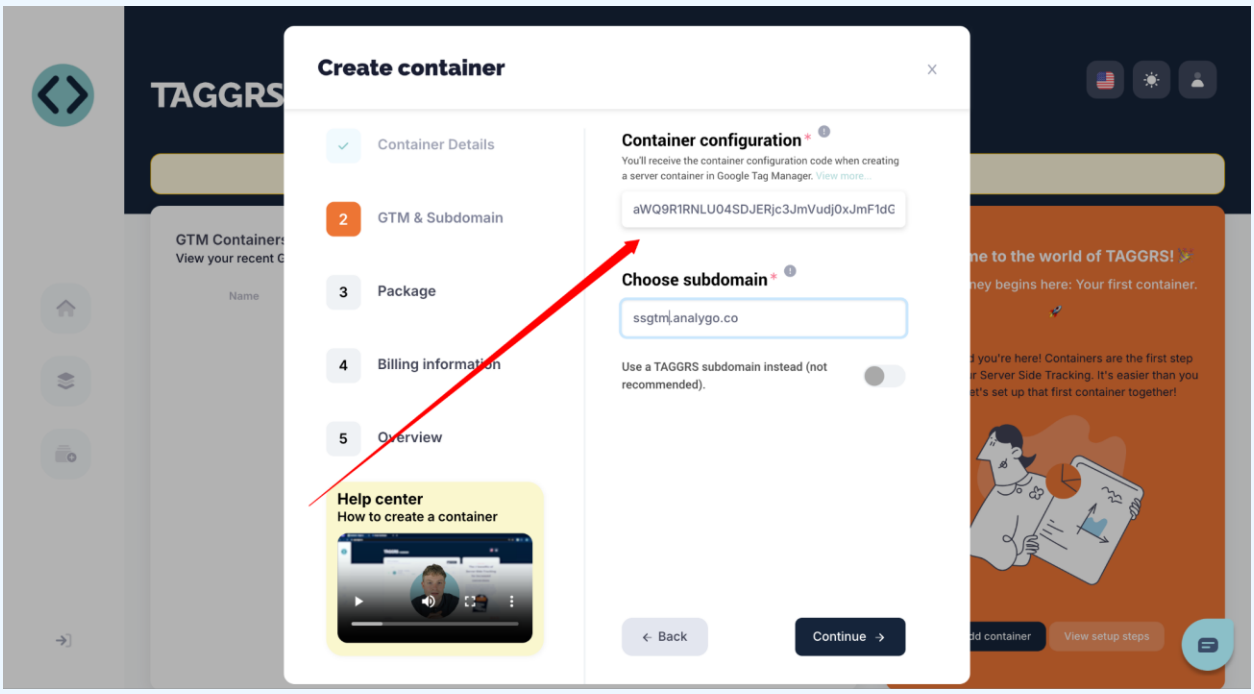


Go back to GTM and copy the ‘Container Config’.

Step 3: add a custom domain

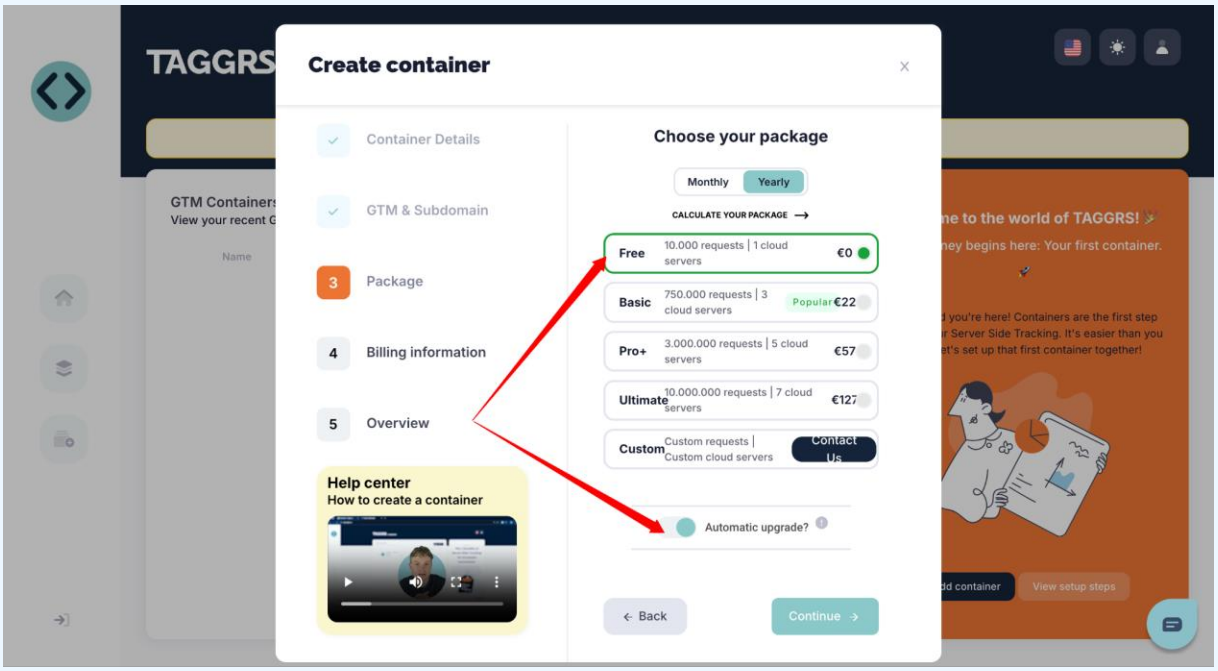


Paste it below ‘Container Configuration’.

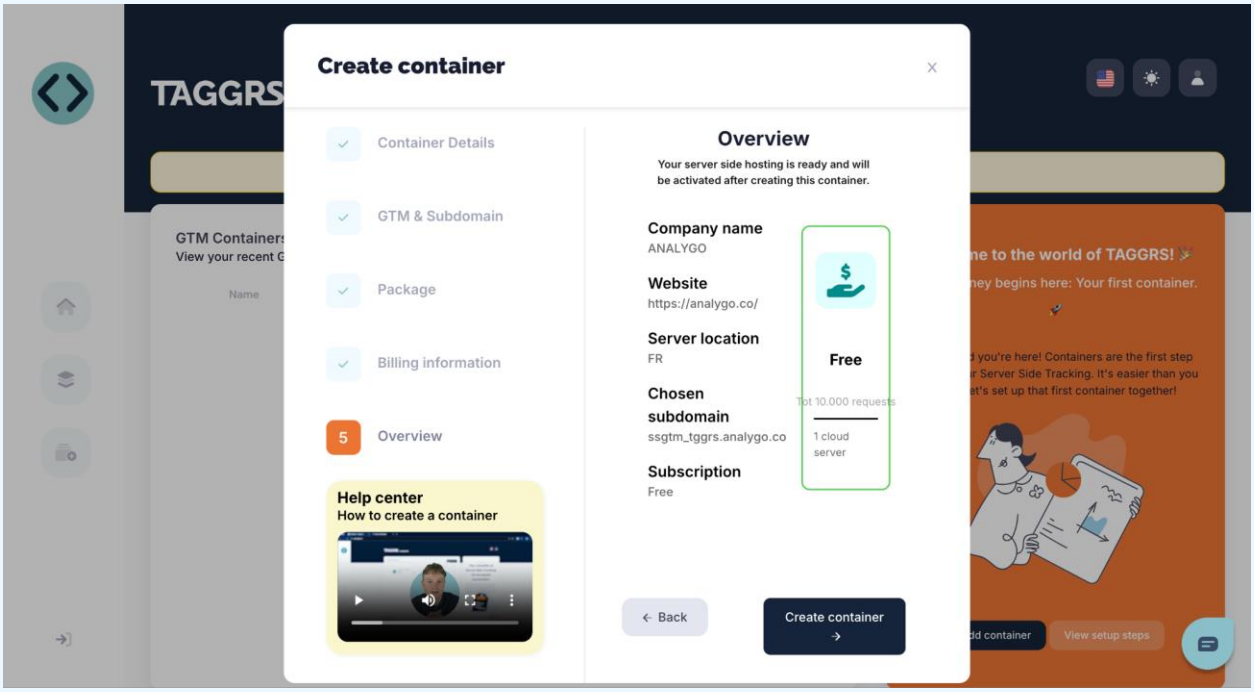


Step 4: select a subscription plan

If you have a tiny website, you can select the free tier. It can manage up to 10000 request (hits).

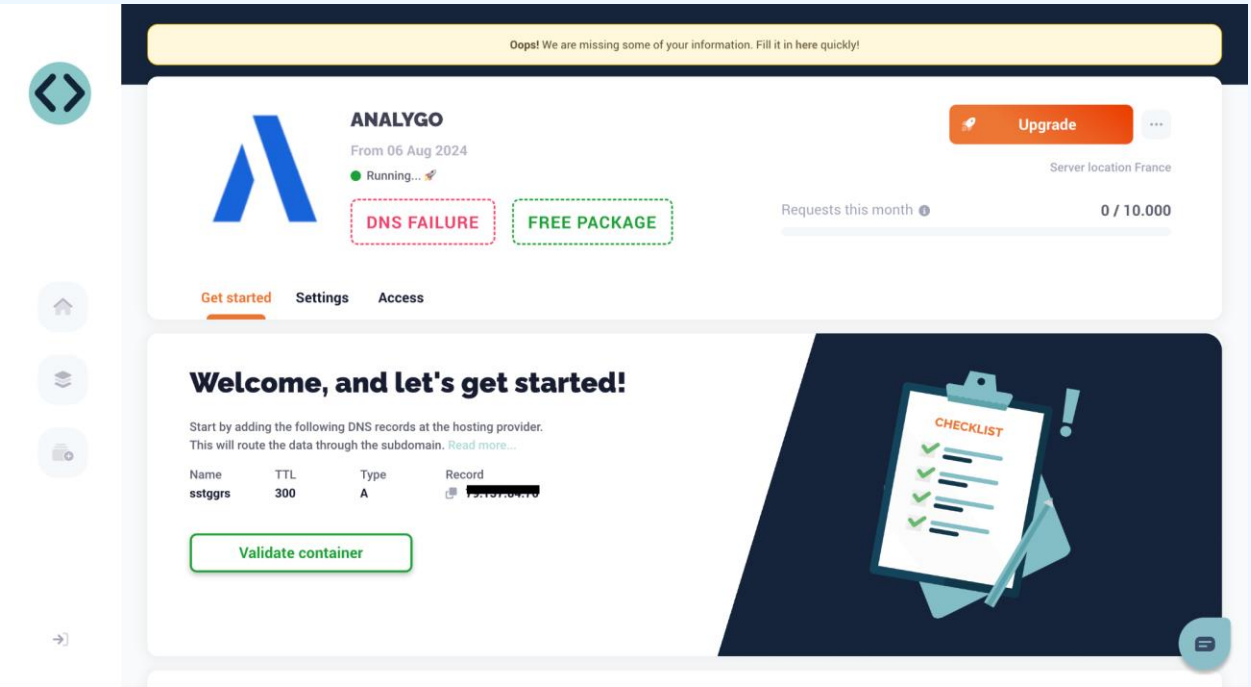


The set-up will autoscale automatically, so don’t worry if you don’t have an estimate for the number of requests.



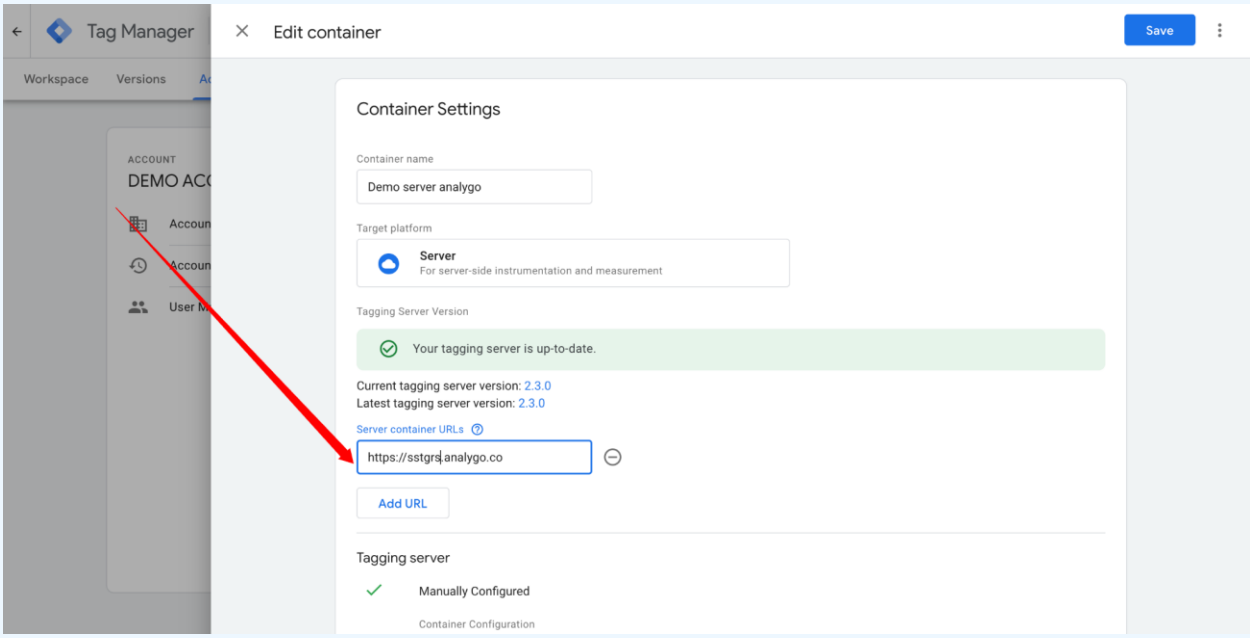
Step 4: add DNS records

After clicking on "Create container," you will have an almost-ready production ready container.

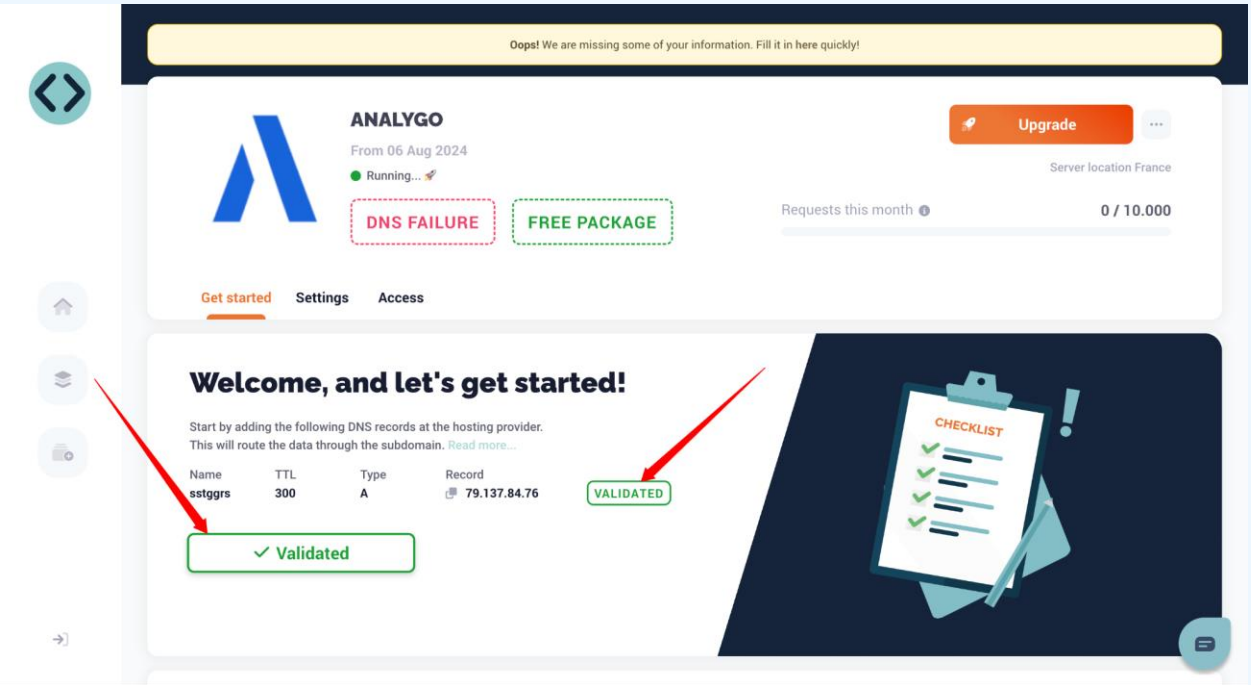


Copy the DNS record information and share it with your developer, or add them yourself if you have the right access.

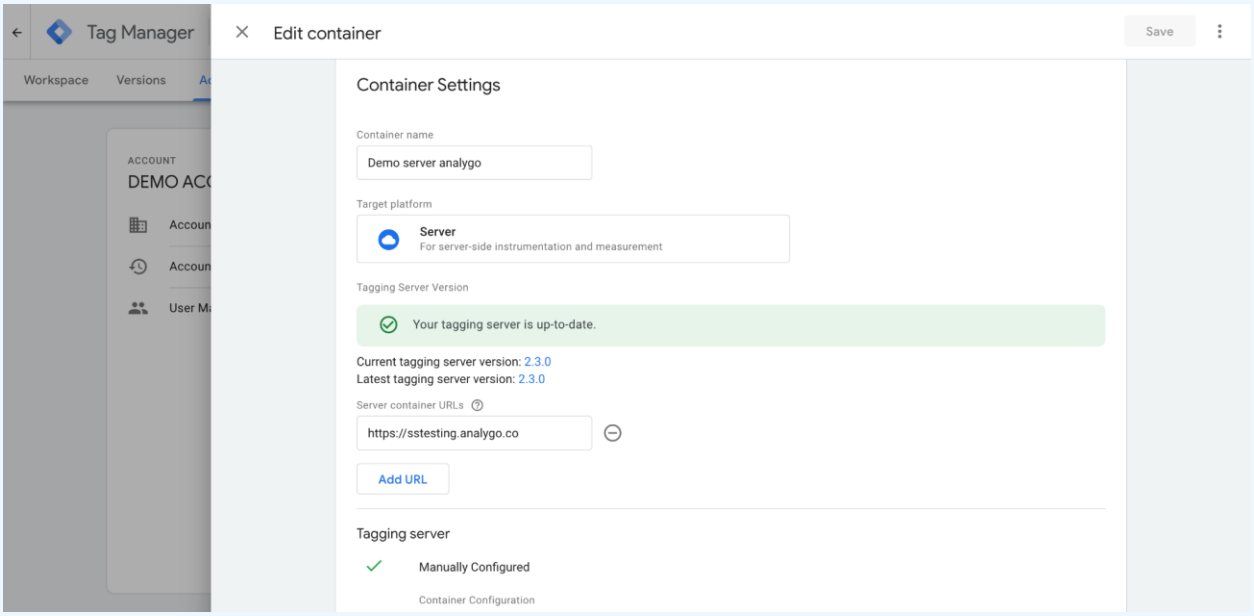
Step 5: add the server URL to your GTM container



Wait a few hours for the mapping to be finished.



Once all is done, you can paste the server URL in your server GTM container by going to ***admin > Container settings > Add URL***.

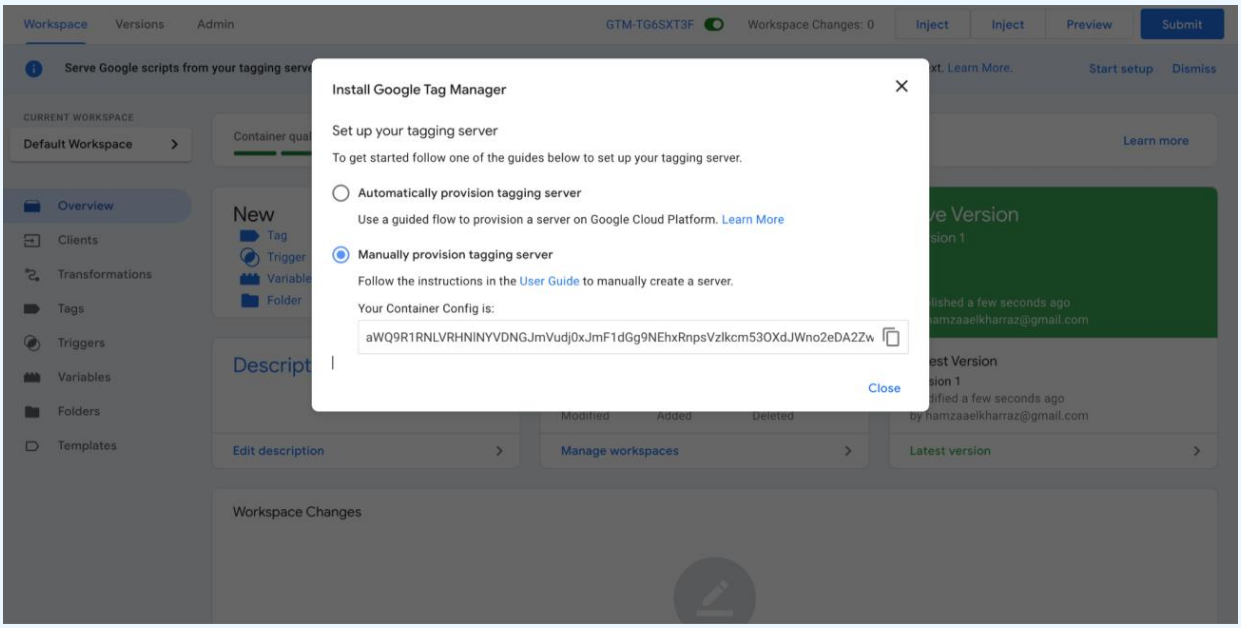


Server-side set up with Addingwell

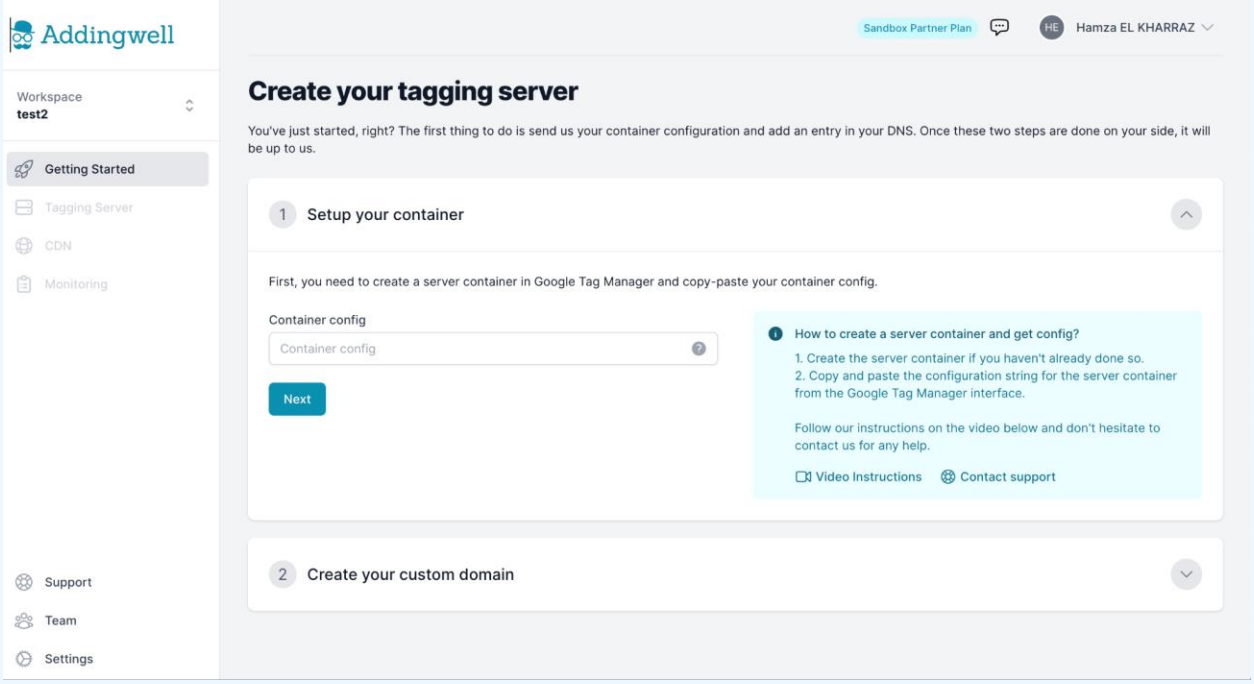
Setting up your tracking server using Addingwell is pretty straightforward. We will start first by setting a GTM server container.

Step 1: manually provision server-container

Create a new server container. Select ‘Manually provision tagging server’. Copy the link.

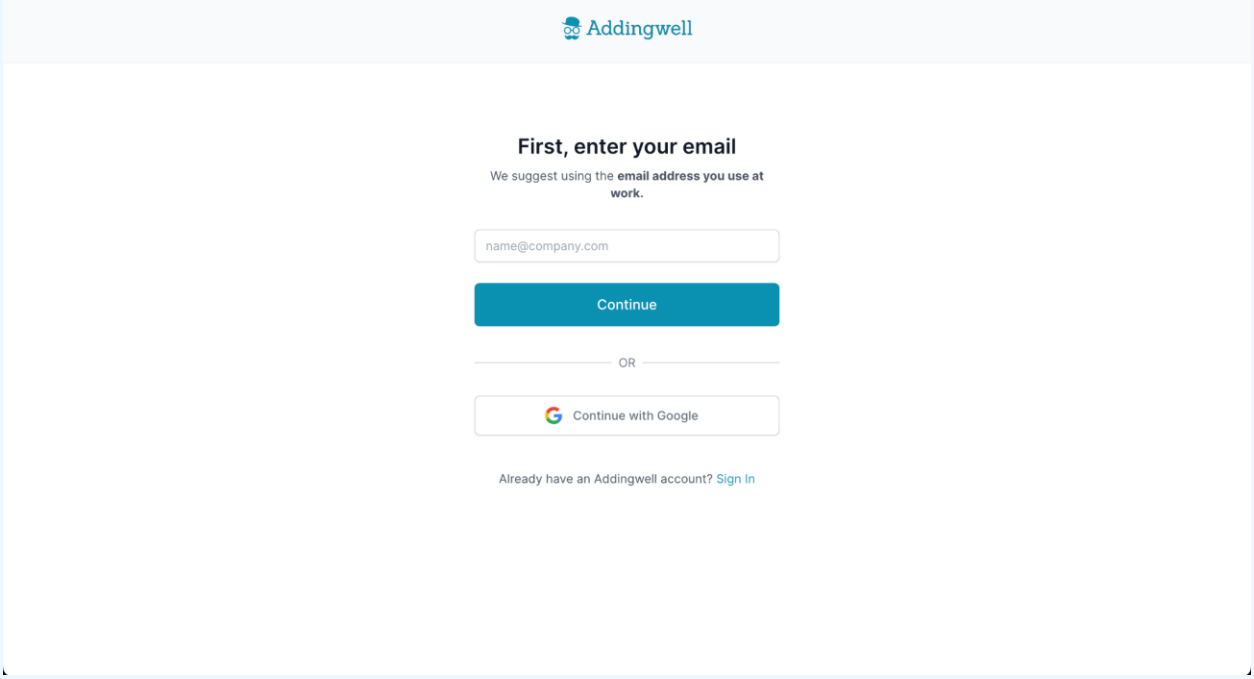


Manually provision the server and keep this tab open.

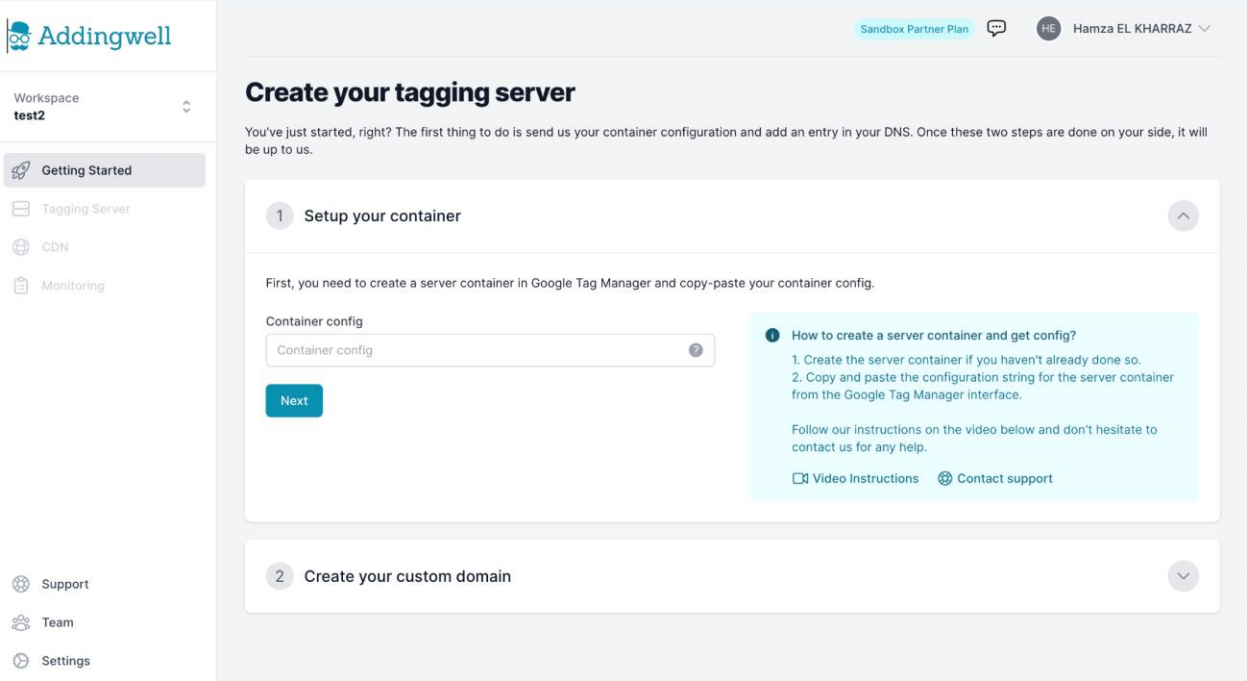


Step 2: create an Addingwell account

Go to Addingwell's [sign-up page] and create a new account.

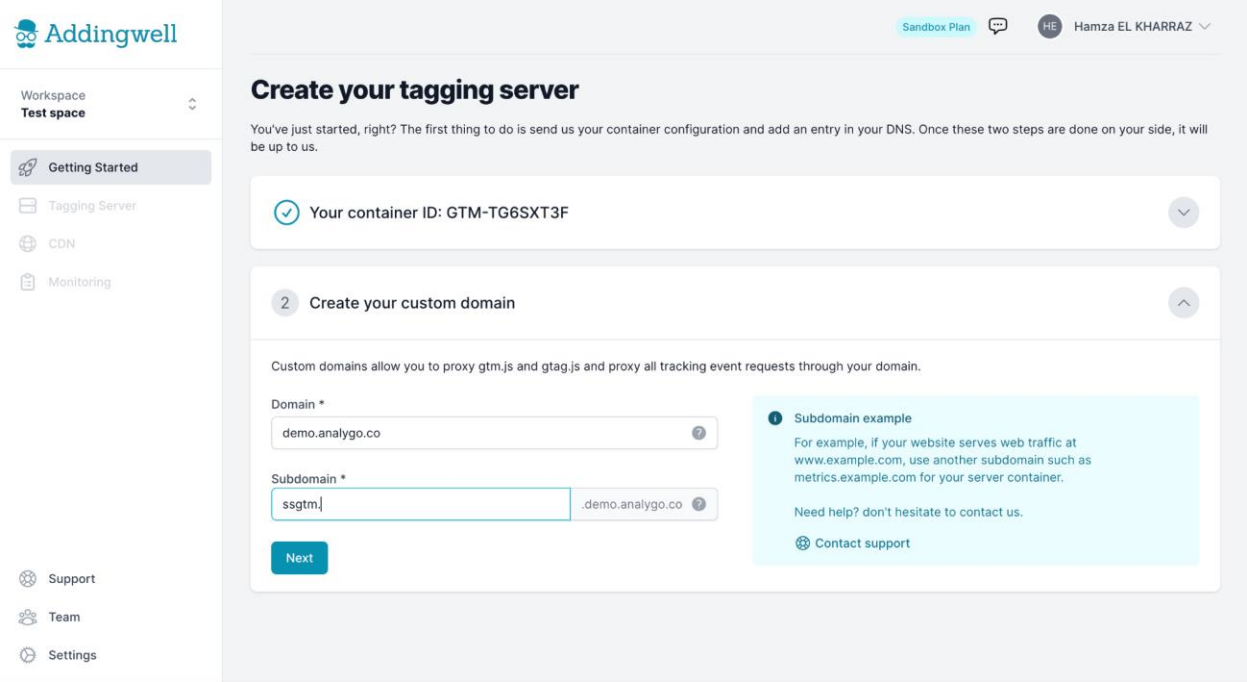


After signing up, you will land on this dashboard. Paste from your GTM below ‘Container config’.



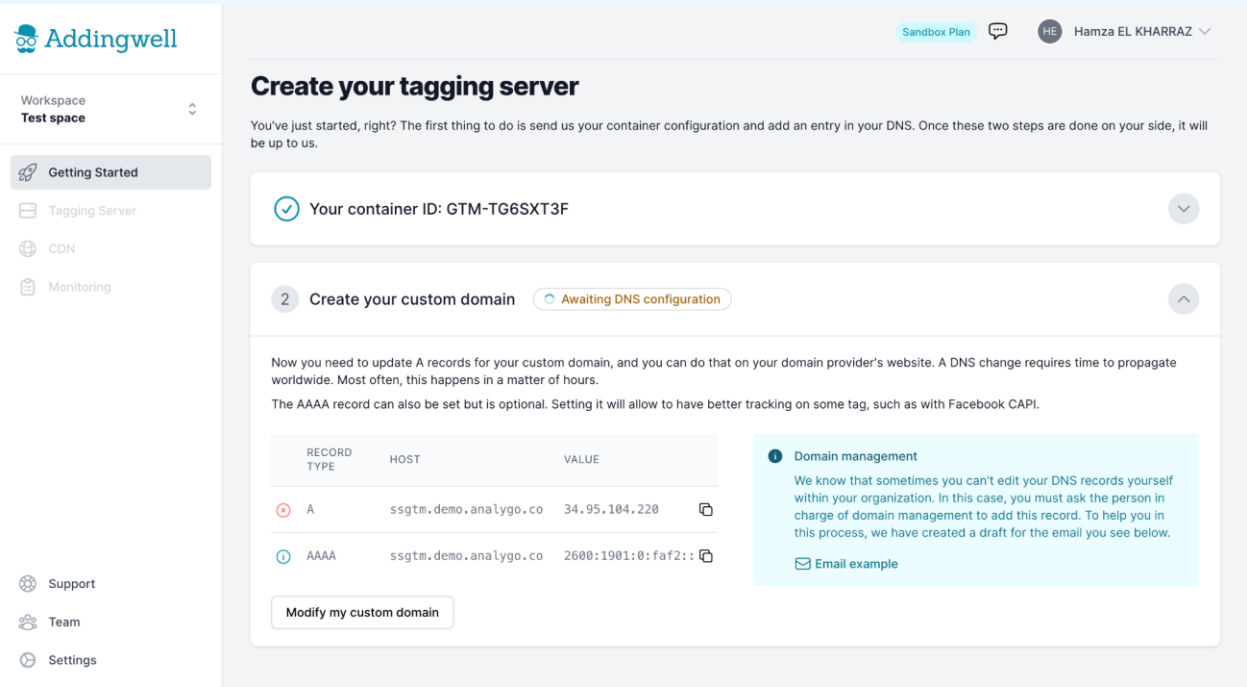
Step 3: add a custom domain

Next, add a custom domain like the one you see below.



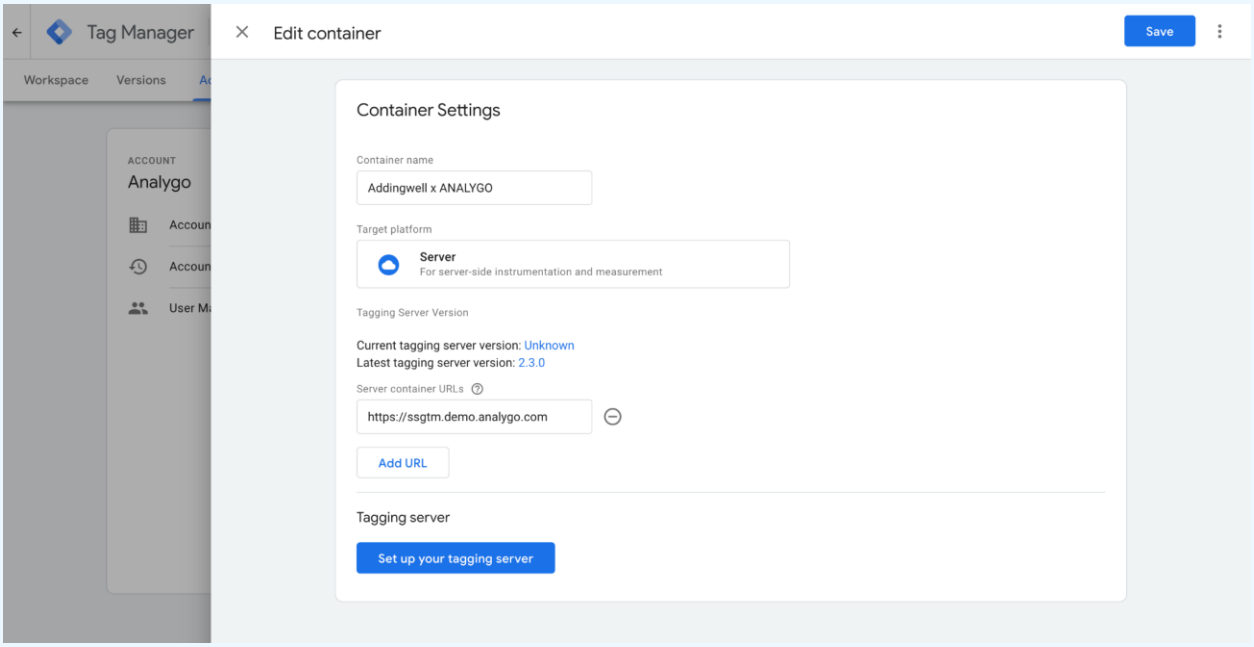
Step 4: add DNS records

Either ask a developer or add the ‘A’ records yourself.

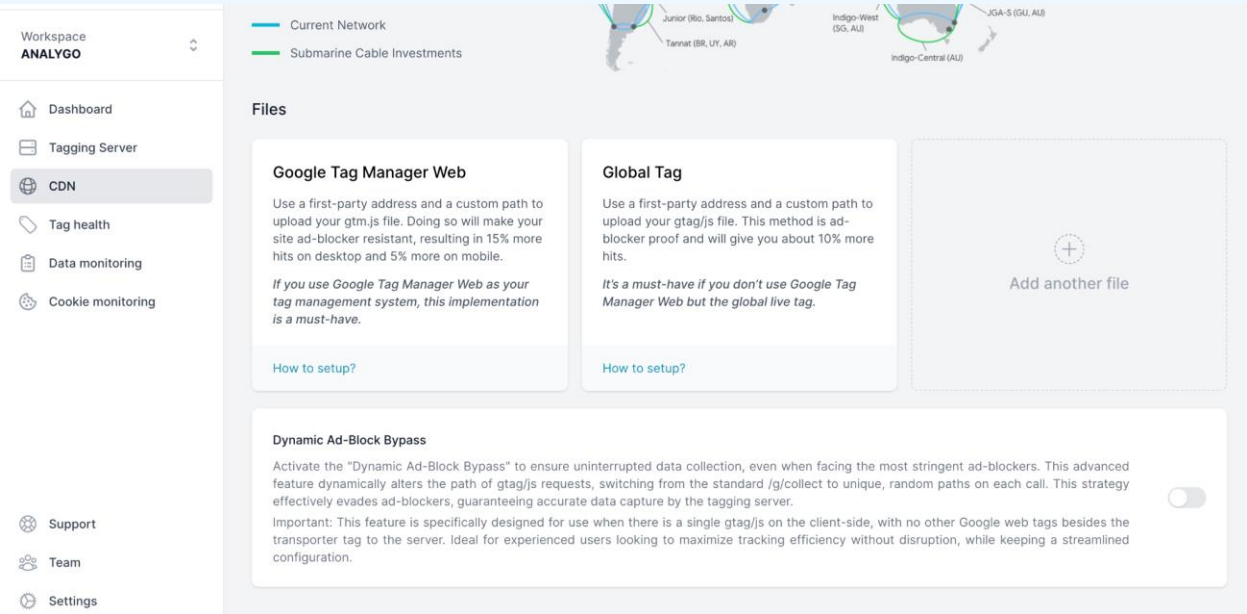


Step 5: add the server URL to your GTM container

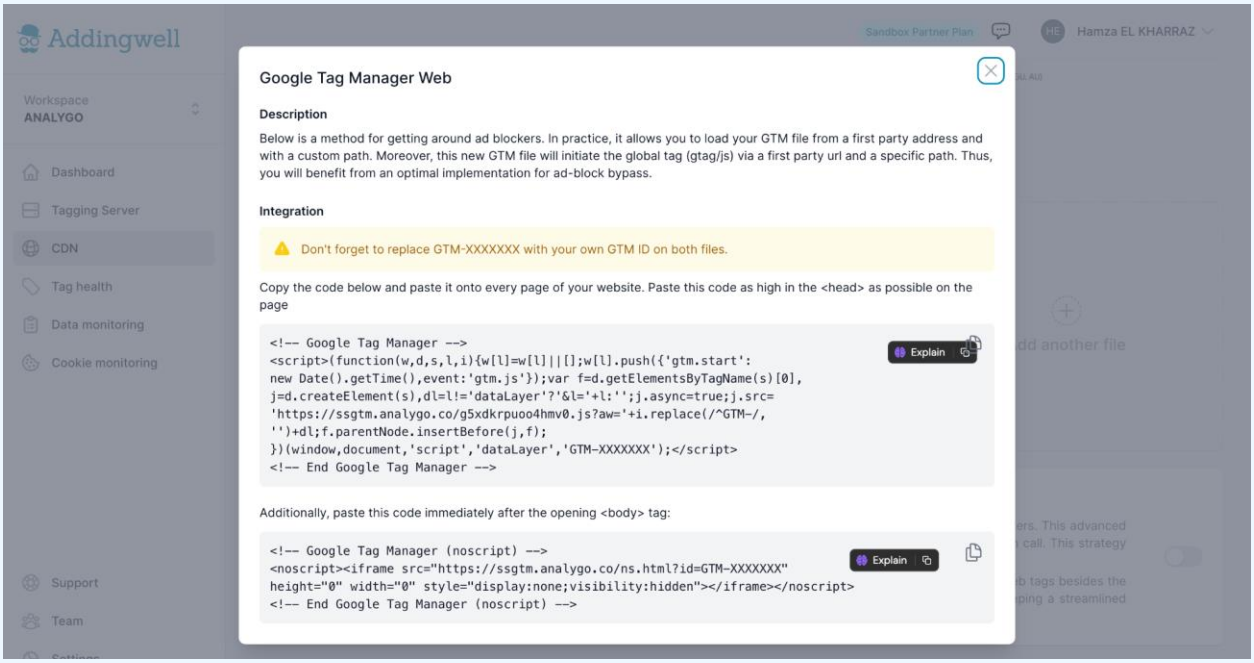
Wait a few hours for the mapping to be finished. Once all is done, you can paste the server URL in your server GTM container by going to admin > Container settings > Add URL.



Step 6: load GTM using your first-party domain



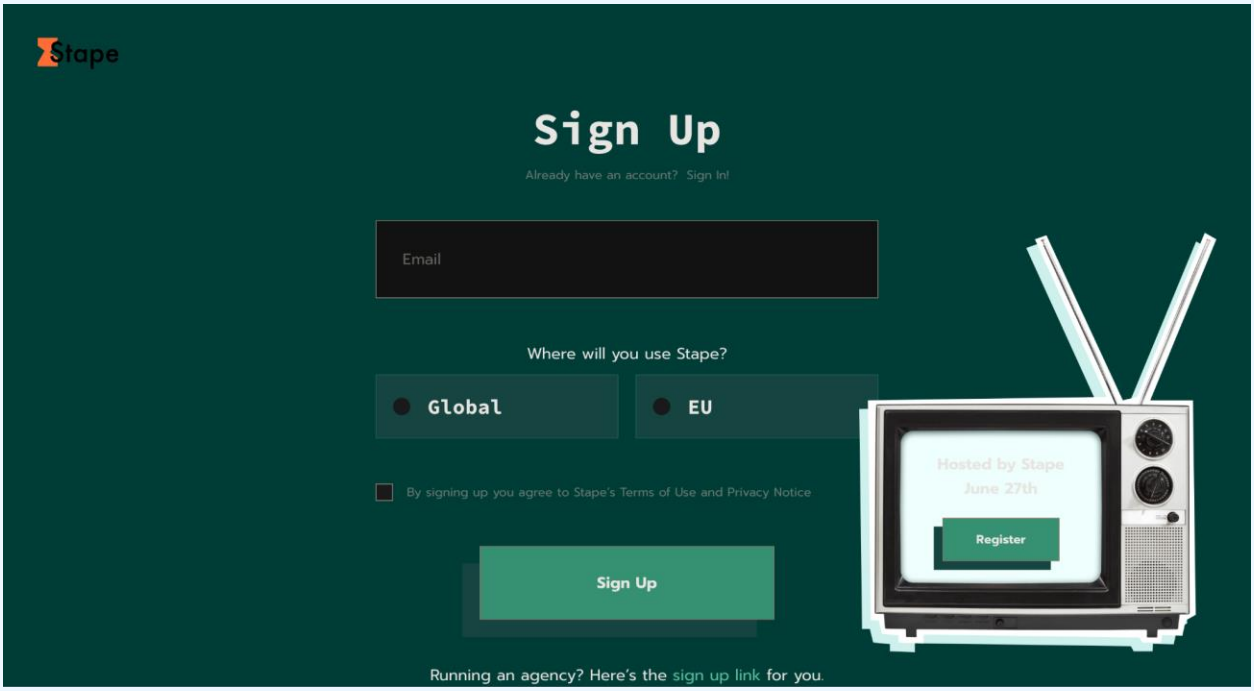
Browse to the “CDN” section. Click on the link under “Google Tag Manager Web”. A new pop-up with appear with detailed instruction how to modify your GTM script



Replace your GTM script with the one specified in your dashboard. **Make sure to change the ‘GTM-XXX’ with your actual ID.**

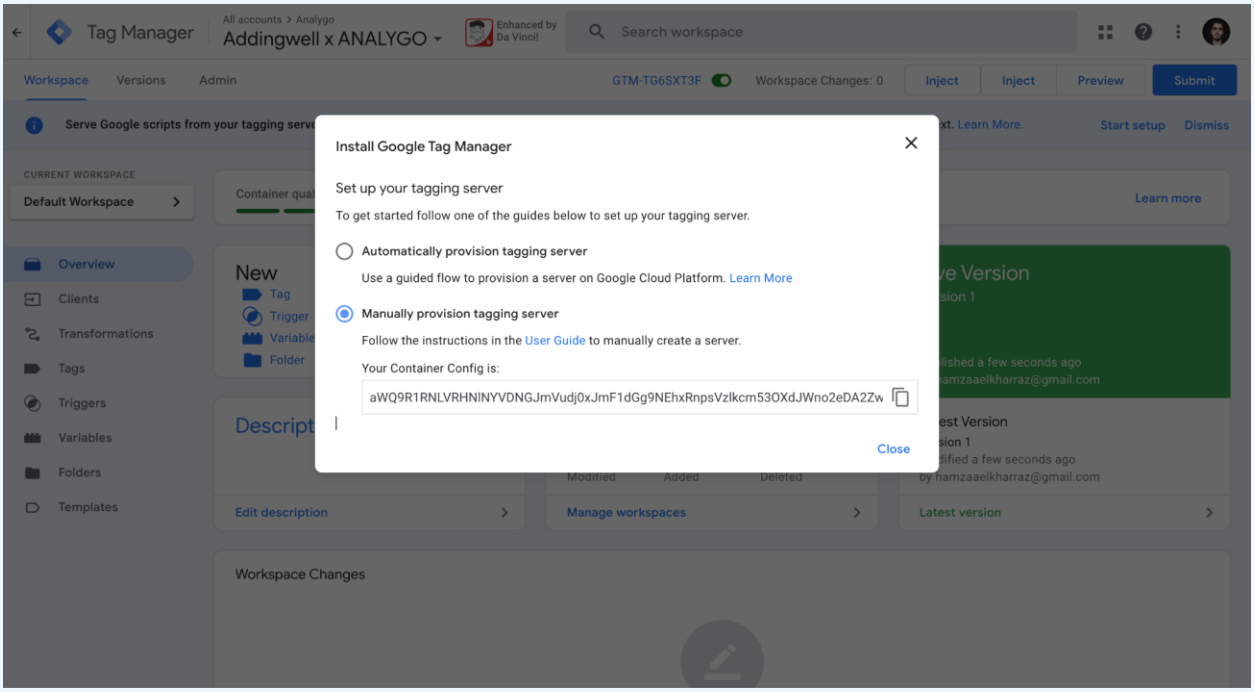
Server-side set up with Stape

Browse to Stape's signup page and start by creating an account. If you are based in the EU region, make sure to check the EU option when creating an account.



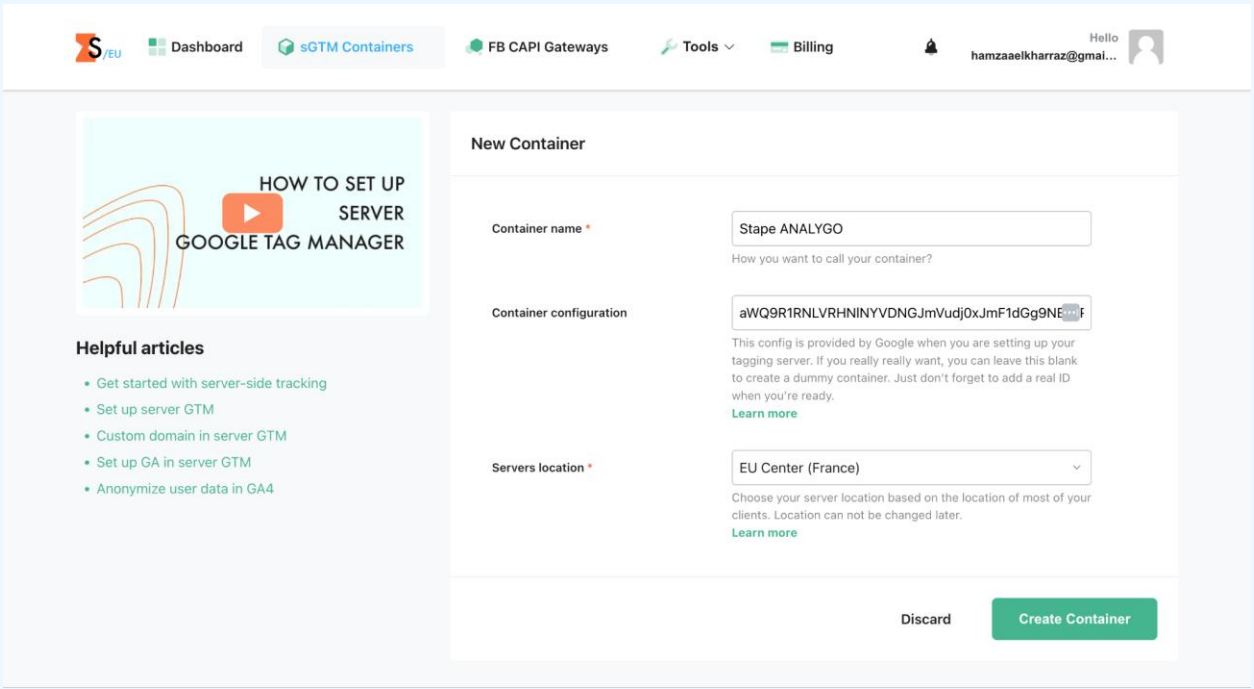
Step 1: manually provision server-container

Select 'Manually provision tagging server' and copy the container configuration link.



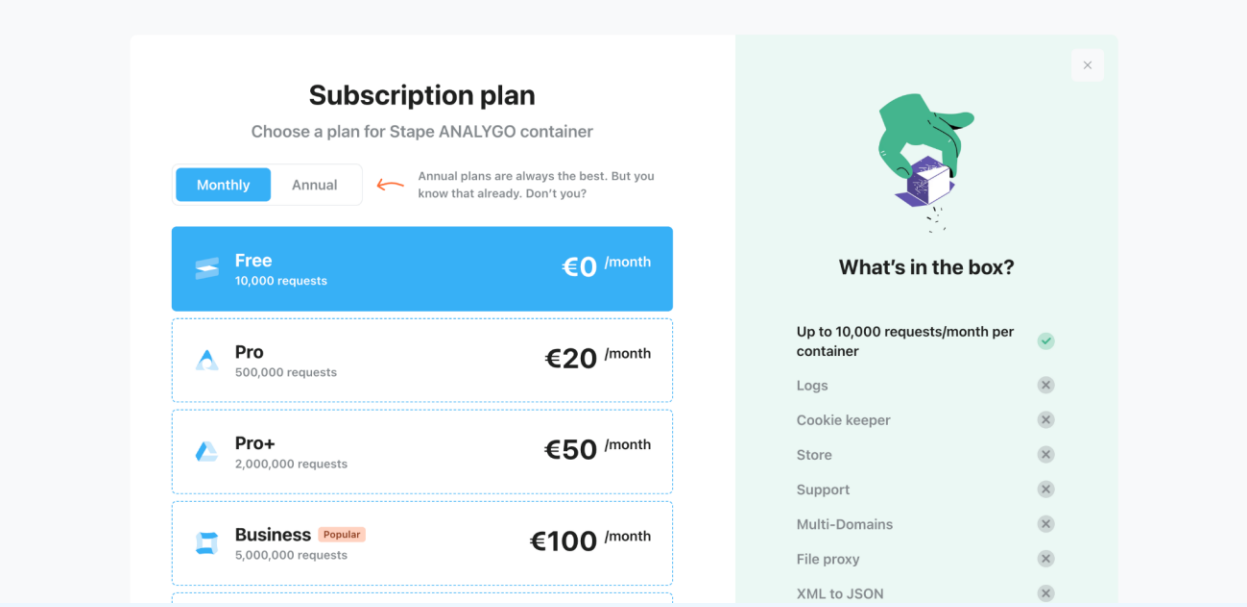
Step 2: Create a new container

In the dashboard section, click on create a container and fill the form to create a new one. Make sure to paste the configuration string in ‘Container Configuration’.



Step 3: select a subscription plan

Select a plan based on the number of your monthly hits (aka requests). Don’t worry if you don’t have a clear estimate. There is an option below to autoscale your set-up if the plan you selected is not sufficient.

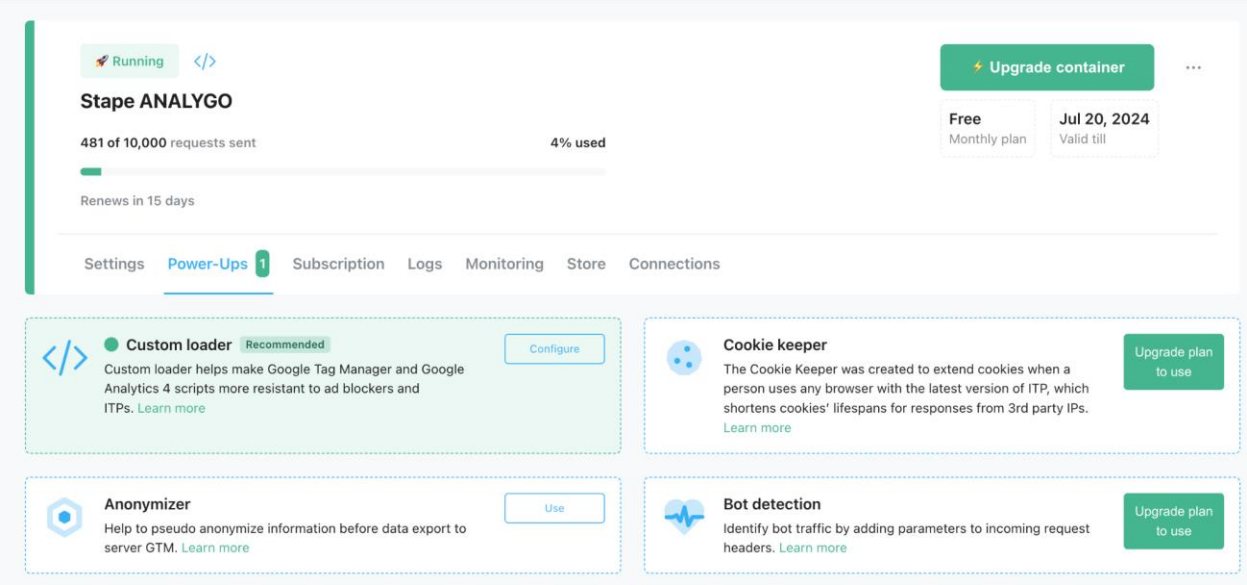


If you have a (very) tiny website, the free tier, which can handle up to 10,000 requests, can cover your needs. Complete the set-up by adding your billing information.

Step 4: add a custom domain

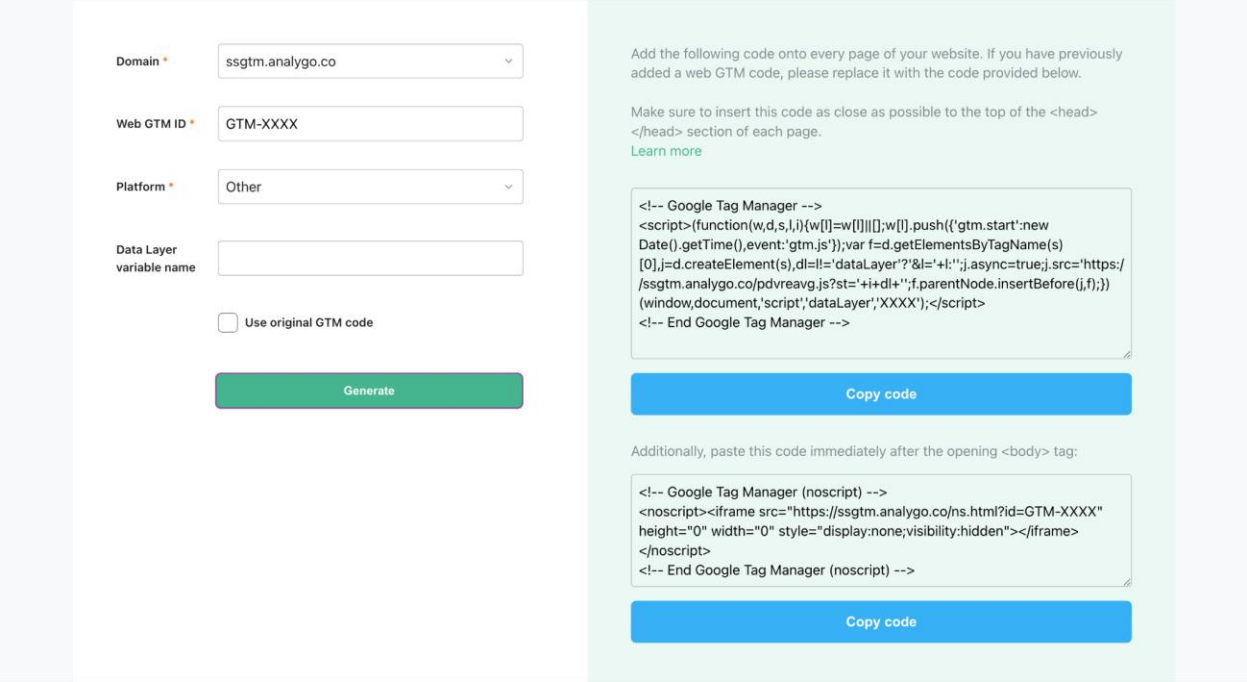
Next we will need to map a custom domain. Look for and click on ‘Add custom domain’. Make sure to add unique URL for your tagging server. Wait for a few hours until the domain verification is completed.

Step 5: load GTM using your first-party domain

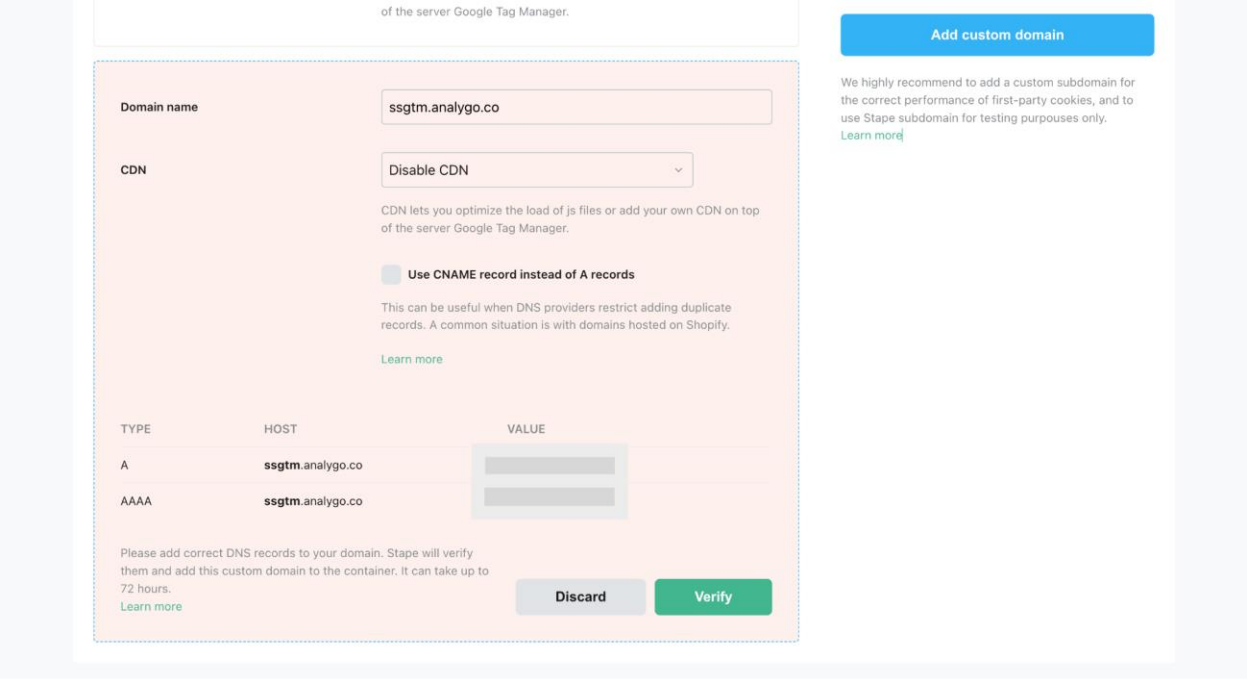


Under the “Power-ups” section. Click on “Configure” next to “Custom loader”. Scroll to the bottom and take the time to fill the data about your container. Once you are done, click on ‘Generate’.

You will get a new GTM installation code that you should replace with the older one.



Replace your GTM script with the one specified in your dashboard. **Make sure to change the ‘GTM-XXX’ with your actual ID.**



Main takeaways

Choosing the right method to deploy your tracking server will depend on a mix of your budget and expertise:

- Maintenance costs

What you will end up paying will ultimately depend on which solution you choose. Using Cloud Run, the costs will vary, deploying a single instance can cost around $45 as a rough estimate.

On the other hand, using Stape or Addingwell, the costs are very straightforward: you pay for what you use.

- Hiring expertise

If you use Cloud Run, you need someone who knows how to work with Google Cloud Platform. On the other hand, vendors like Addingwell or Stape can help you get started and provide assistance when you get stuck.

Note for agency owners:

As an agency owner doing multiple server-side implementations per month, I find that using vendors like Stape or Addingwell is much more convenient, as it reduces the time I have to spend on each individual project.

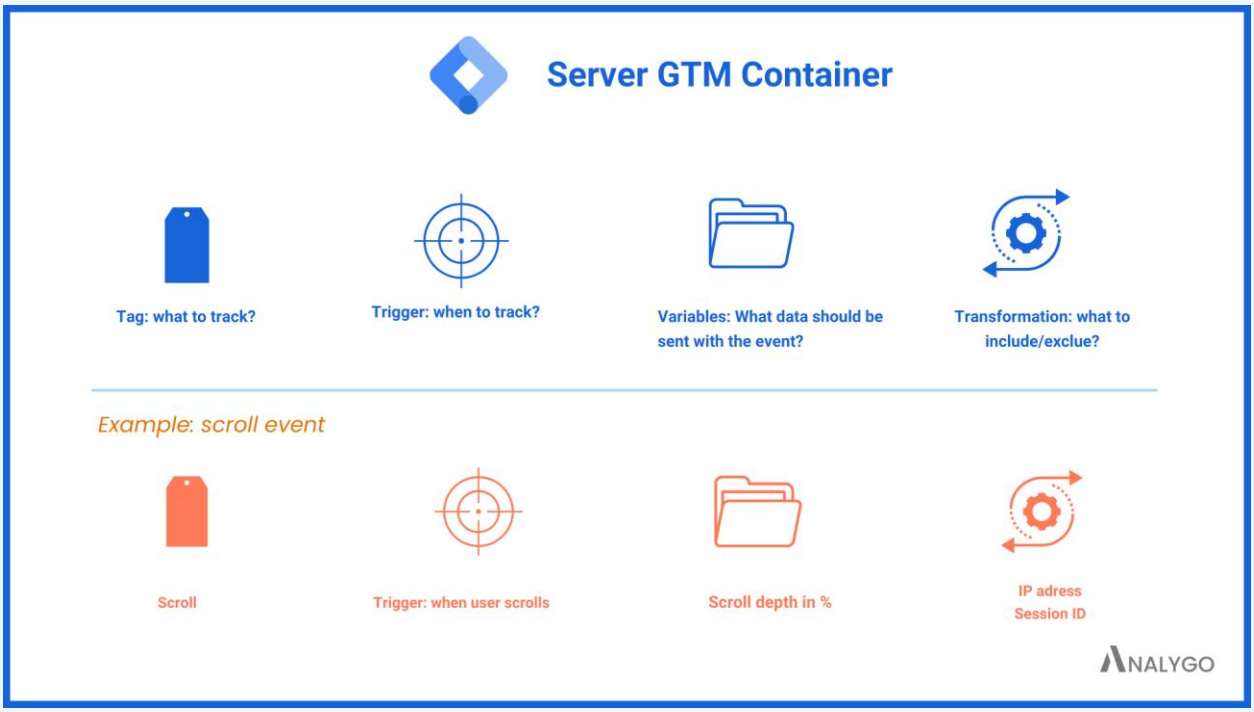
If you are an agency or a freelancer assisting a client, then going with a vendor instead of Cloud Run might make more sense, unless the client is a large business with a team that understands the Google Cloud Platform.

Part 4: Setting up your first tags

"What gets measured gets managed."

Peter Drucker

After all the talk about servers, requests and other server jargon, it's time to get back to familiar territory. **Tags**, **triggers**, and **variables** are all familiar concepts to anyone who's done any kind of tracking using Google Tag Manager.



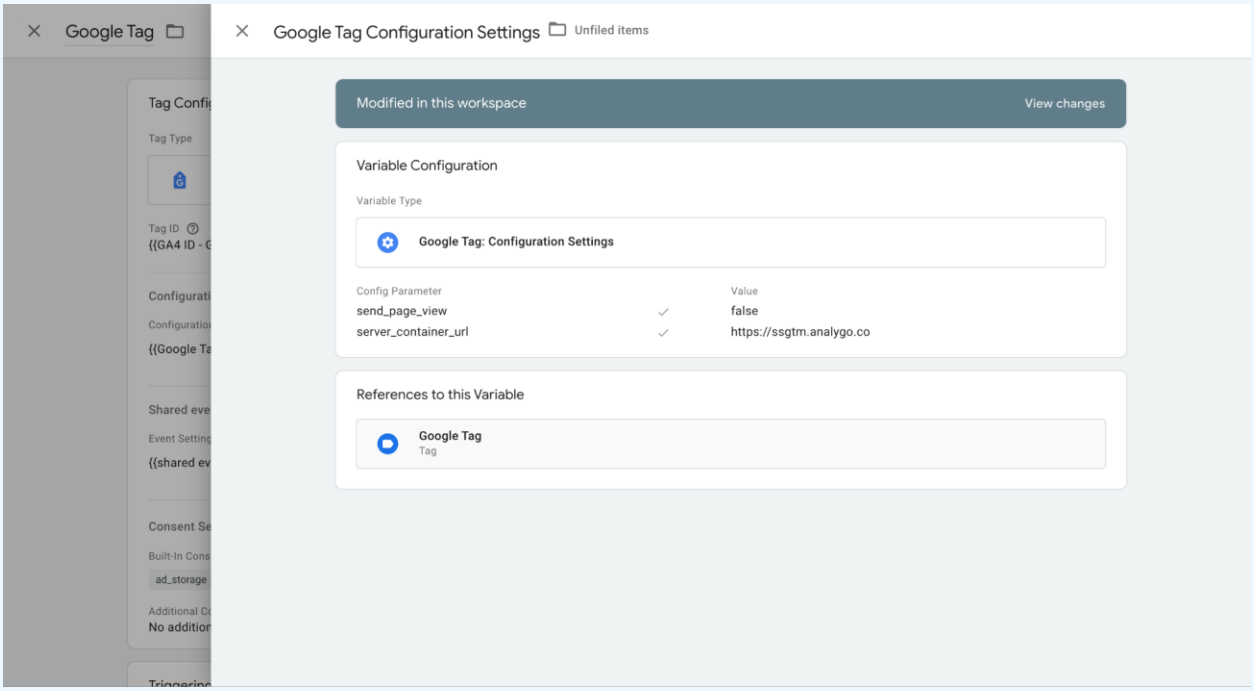
The building blocks of a GTM server container

Transformations are the only part you may not be familiar with. Remember when we talked about how server-side tracking will give you more control? Transformation is what really makes this possible.

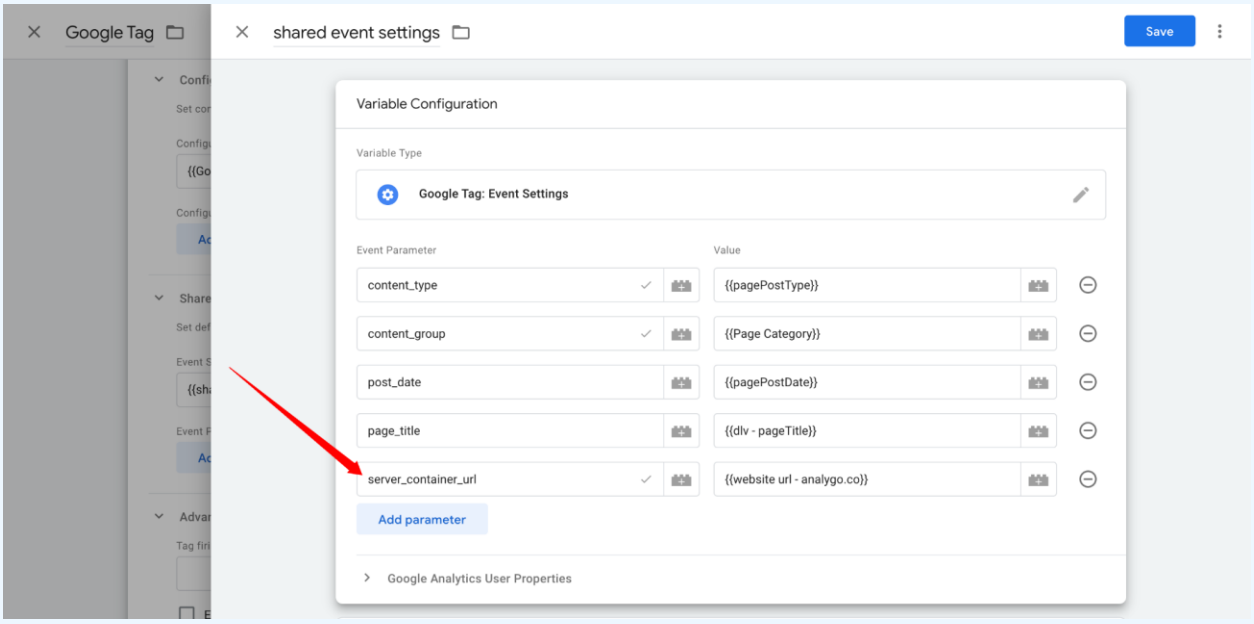
Using the example in the image above, suppose you've got a scroll tag, but you don't want to send the IP address because your legal team told you it may not be a good idea. Using transformations, you can explicitly block the IP address from being sent to the vendors.

Example of setting up a GA4 tag

In the web container, make sure that your server tracking URL is added to GTM's configuration tag.



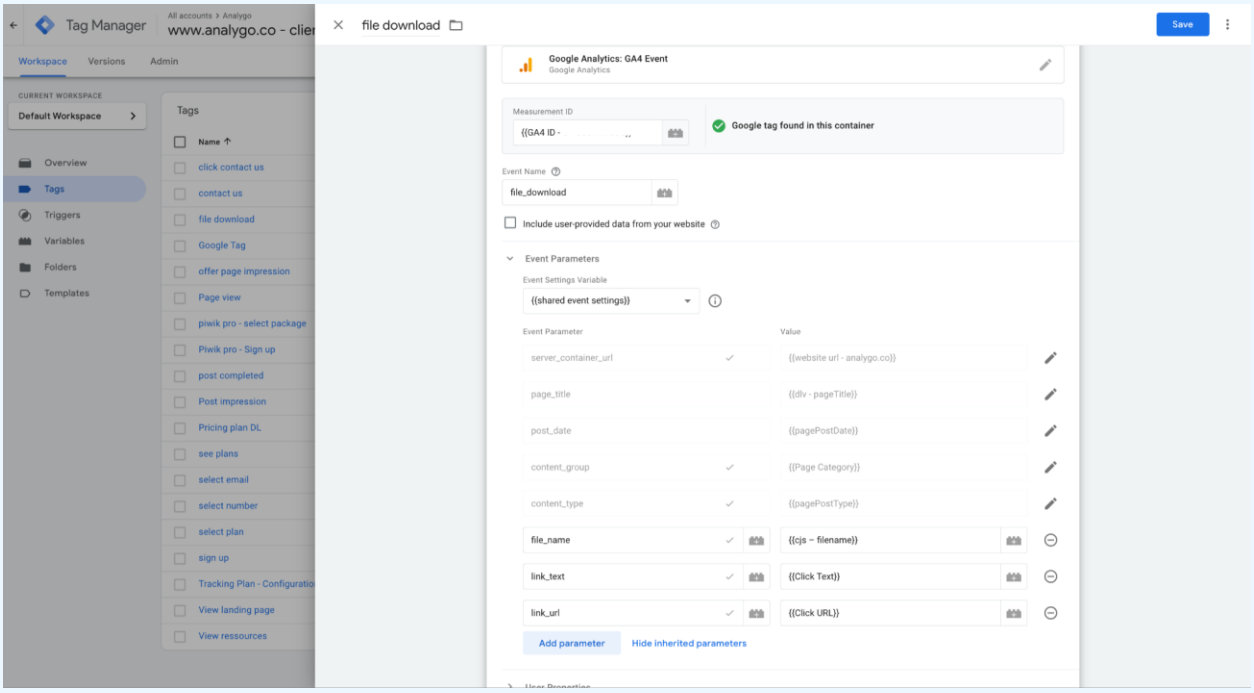
Repeat the process in the shared event settings.



Web event tag set up

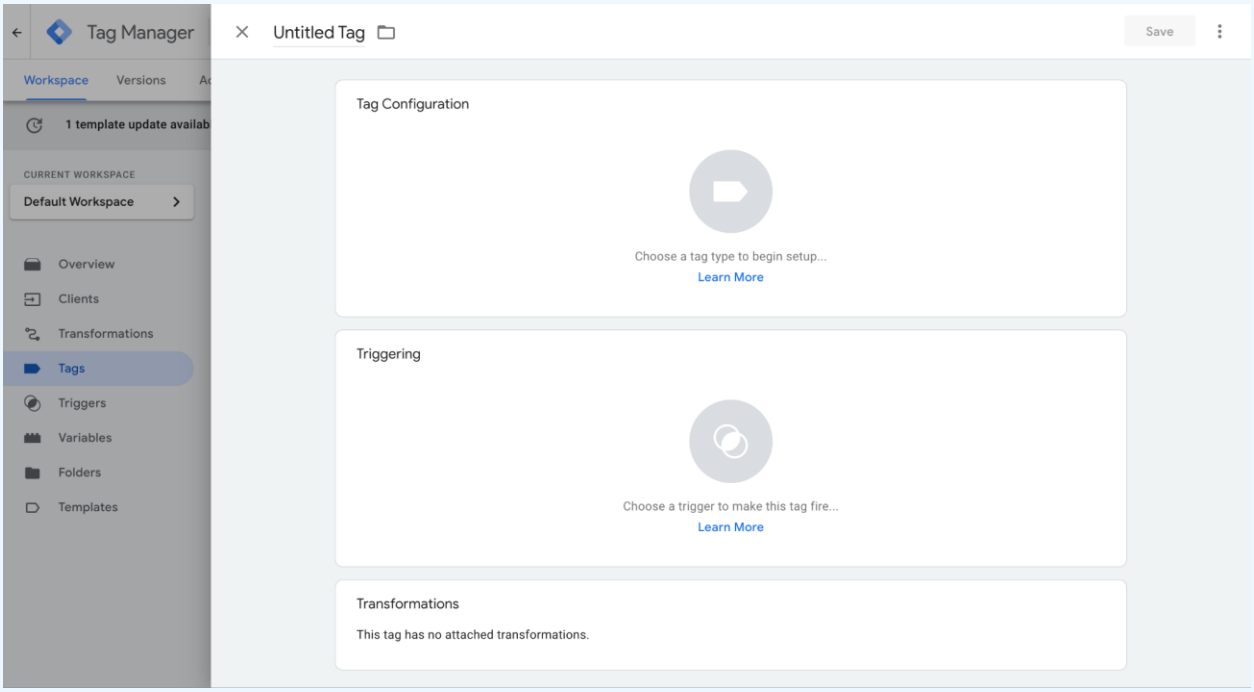
Do not pay attention to the other parameters for now.

Now let’s let’s move to our first event. I want to track the number of file downloads. The e-book you are reading now is one of the files I’m tracking on my website using the tag below.

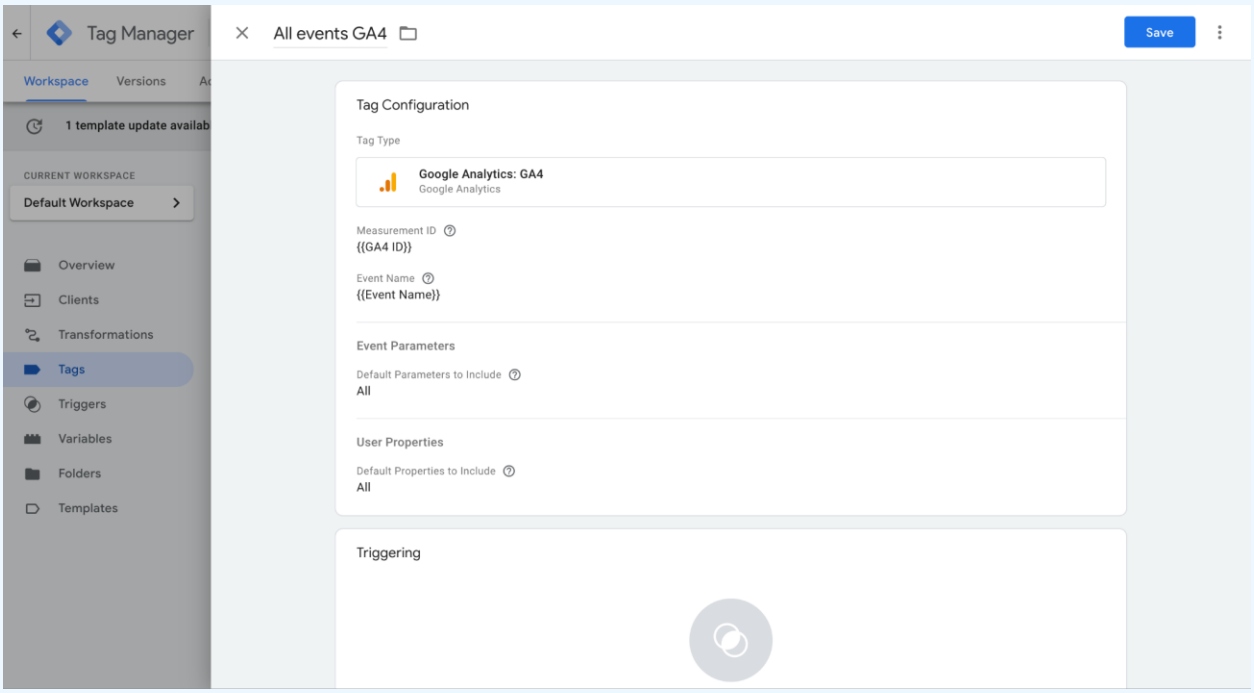


Server tag set up

On my server GTM container, I will create a new tag.



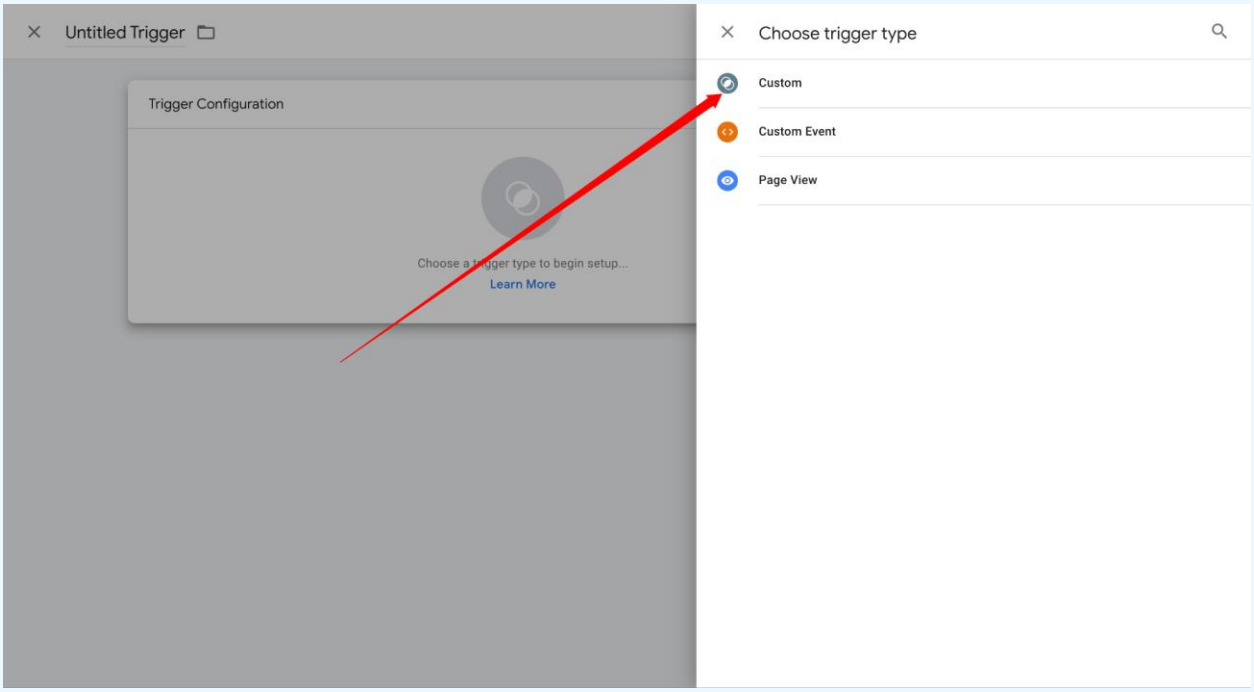
Next, select GA4 event tag template and add things like your GA4 ID. Notice that I named my tag ‘All events GA4’.



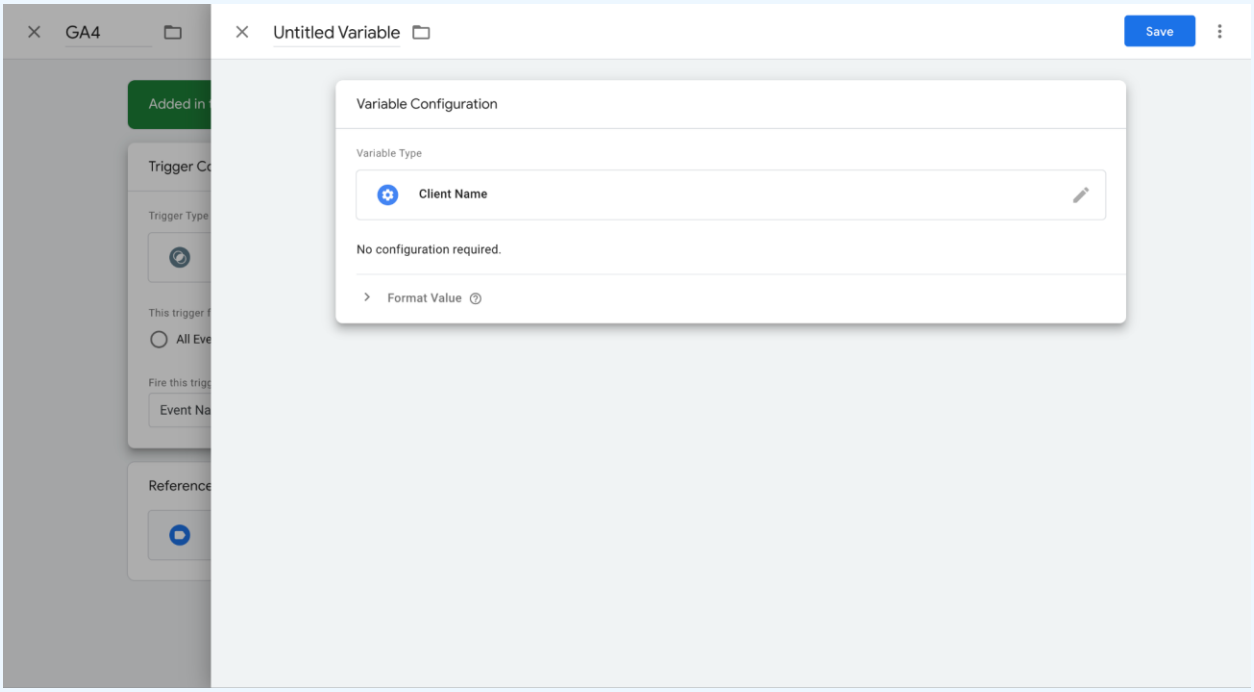
Since the server container acts as a middleman between GTM in the browser and a vendor, it merely receives events, modifies them, and then forwards them to the vendor.

Why am I telling you this? **Because we only need one tag**. The “{{Event name}}” variable I placed under the name of the event in the screenshot will take care of collecting all events dynamically as long as I add the events I want in the trigger.

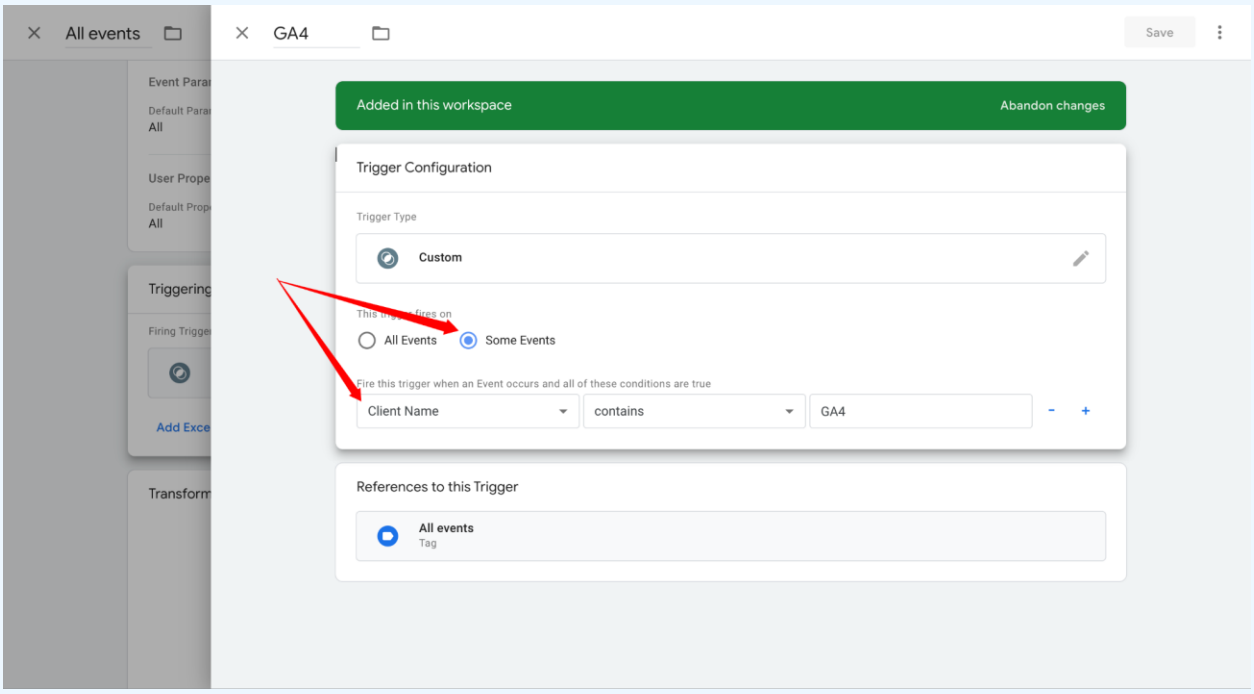
Now, let’s add a new trigger and add a **custom trigger**.



Make sure to select the tag fires on ‘Some events’. Add, a new ‘Built-in variable’ and search for ‘Client Name’.

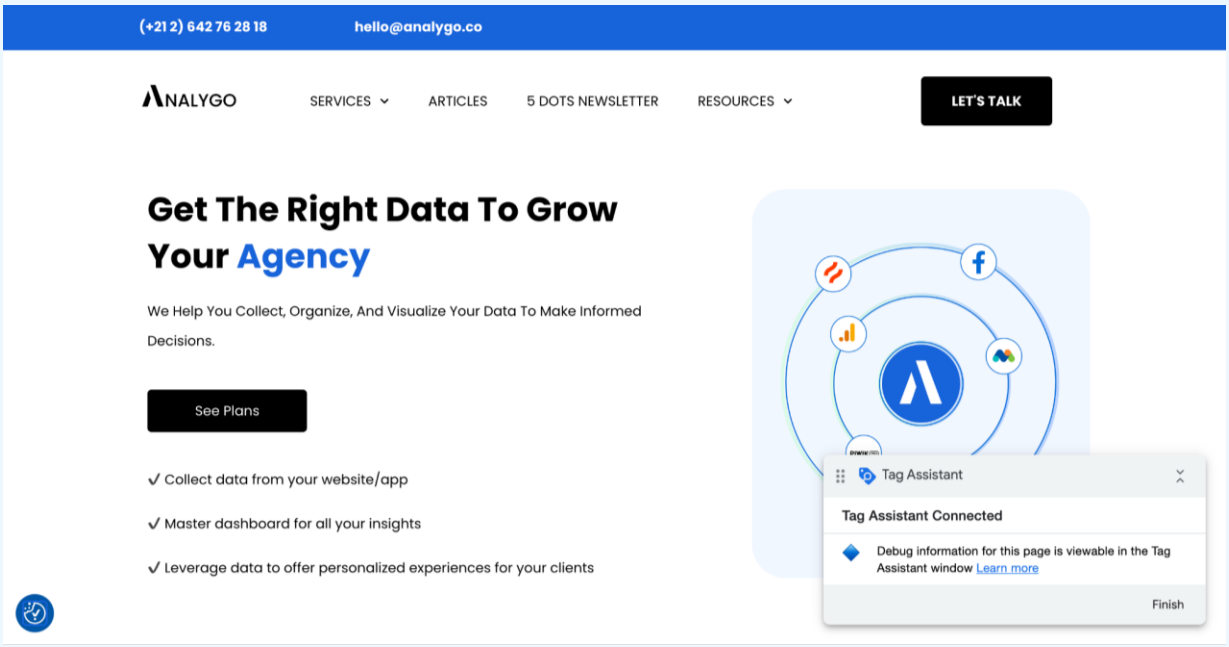


We only want events that can be claimed by GA4 client. In the empty condition field, insert ‘GA4’.



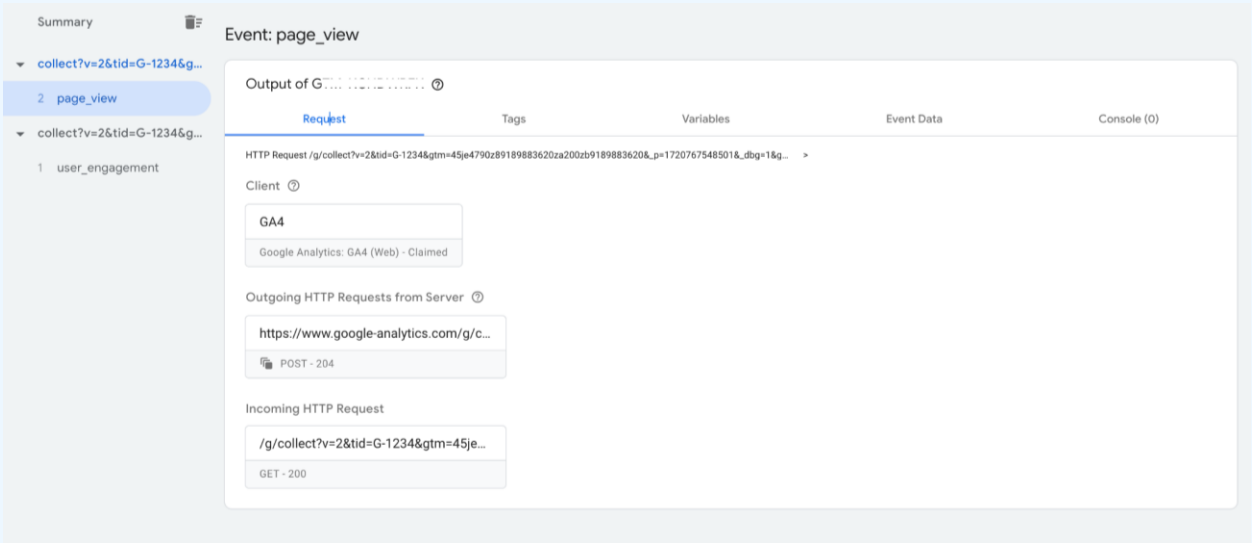
Test the set-up

Now all we need to do is to preview our server container and see the changes we made. You need to open the debug mode of your client container **at the same time.**



Make sure to enable the preview mode in your browser's GTM.

Now, let's take a look at what we get on the server side.

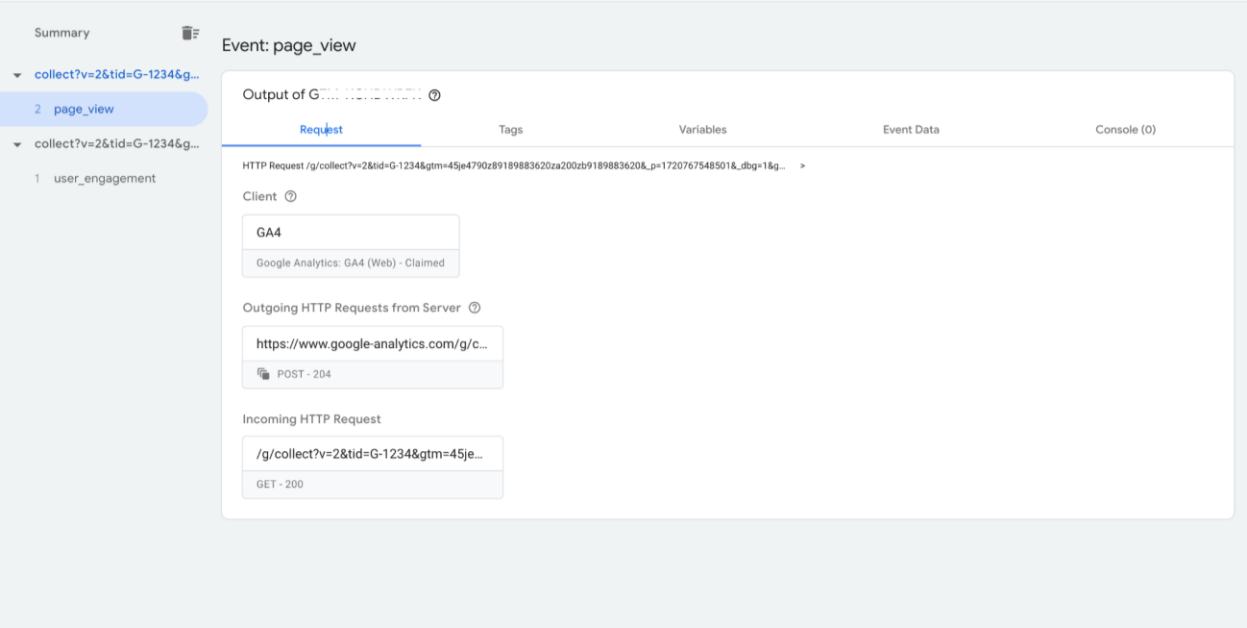


As you can see, we already receive some events from our client container. Now, don't let this screen scare you. It has some familiar tabs like tags from our web container along with some few new tabs:

- Request tab

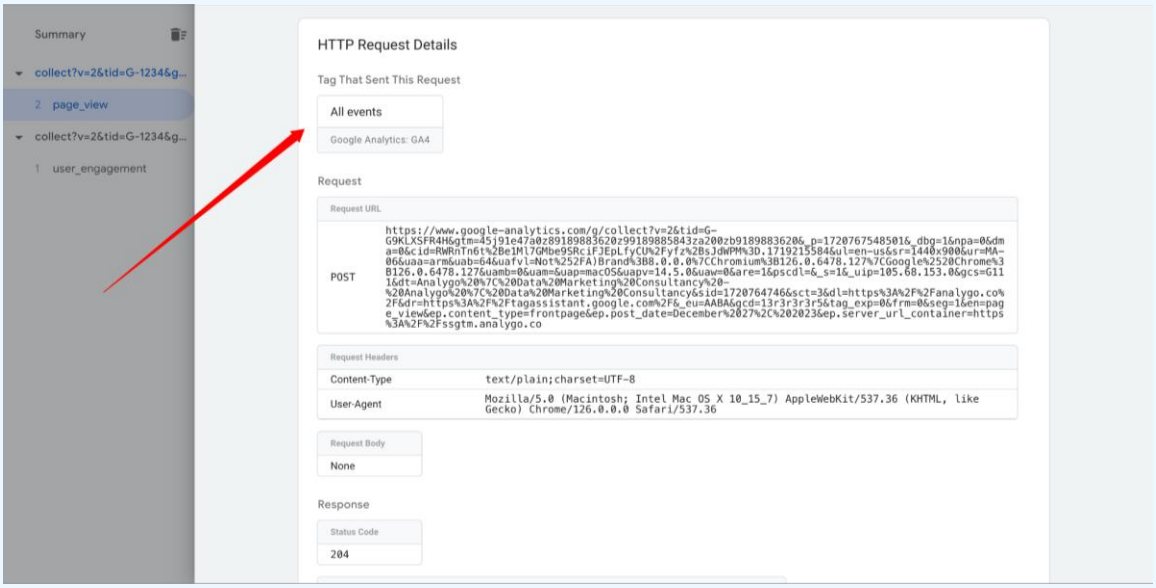
By far, this tab is the most critical to understand the data transfer happening in your server container. By looking at this tab, you will understand:

- Who claimed the request: a client will make the event available after claiming the request (remember the translator analogy?)



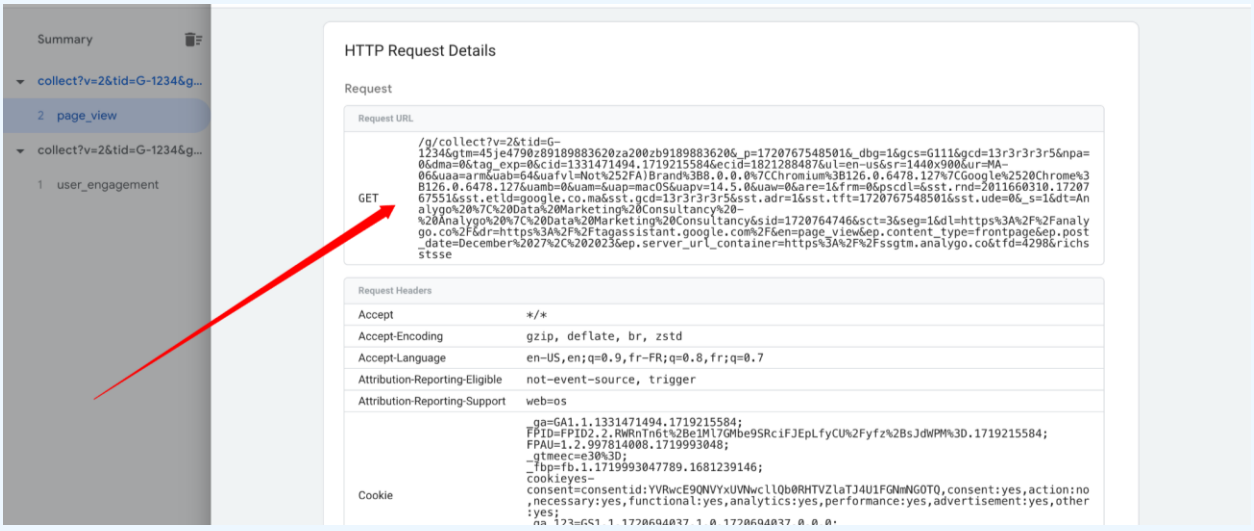
- Outgoing request

This section will display the final destination of your data, in other words, which vendors are receiving the event. In the screenshot below, the GA4 tag 'all events' sends the data to GA4, for example.



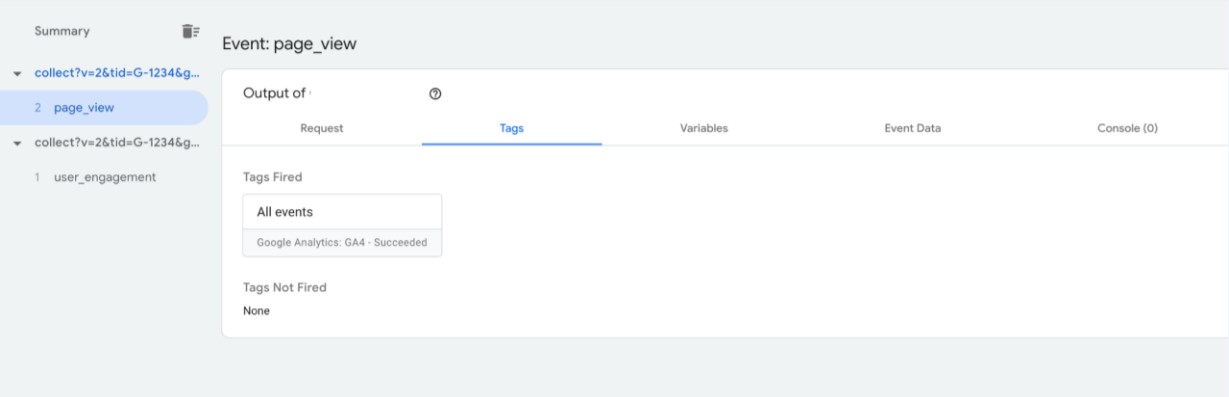
- Incoming request

Here you will know more about the request sent to us by the client container. All the information about the event - from user's IP address to the page URL - can be found here.

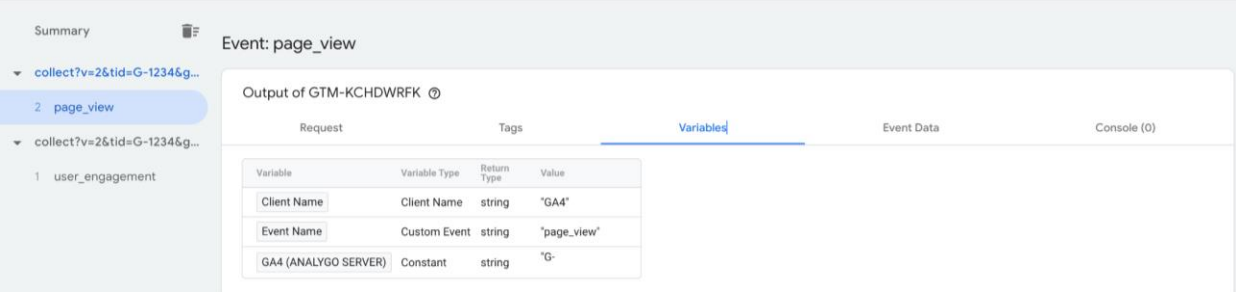


Tags tab

Here you will find all the tags that fired for a specific event like 'page_view'.



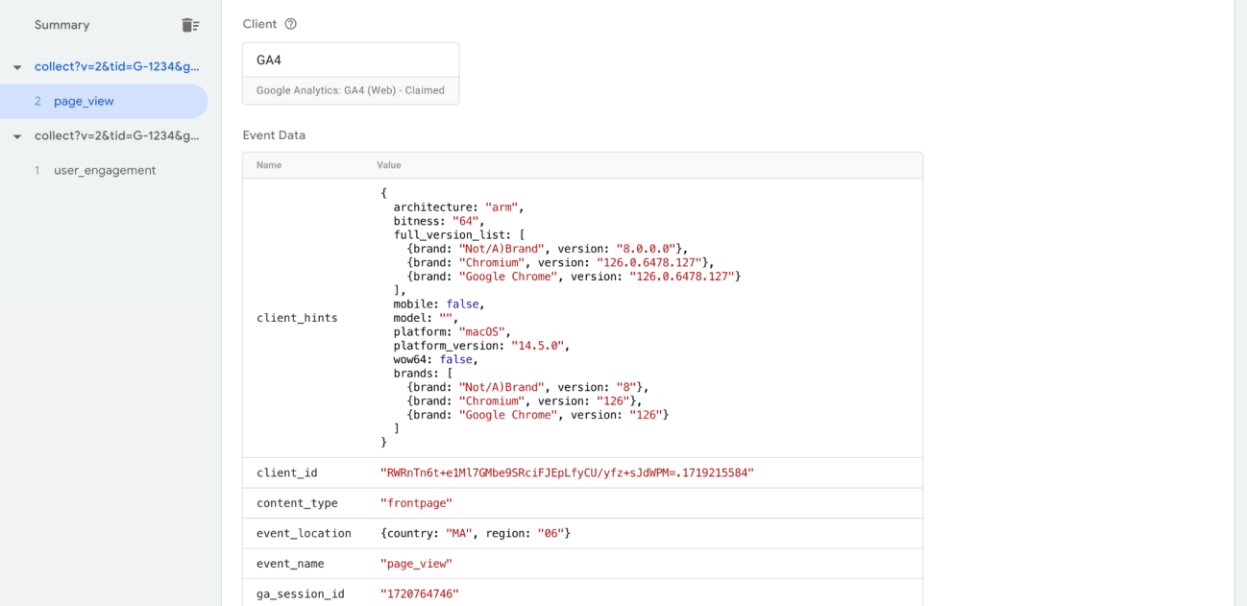
Variables tab



All the variables I create in the server container will be in this section.

Event data tab

This table will give you all the detail for the event claimed by your tag. It answers the questions “what kind of information I’m sending in with this event?”



Console tab

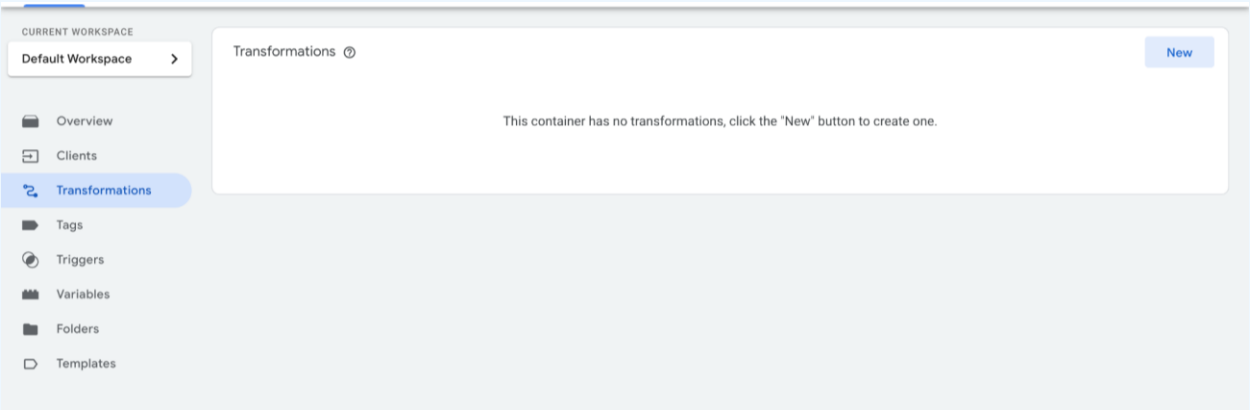
Finally, the errors tab will highlight anything that went off the rails with your tracking. When there is an error, you will see ‘Console (1)’ which means 1 error was found.

Modify events using transformations

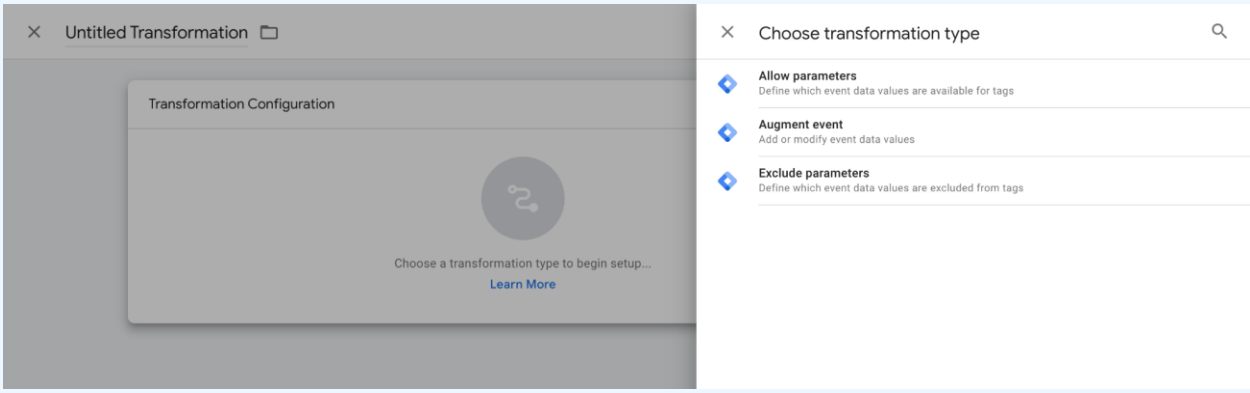
Transformations allow you to exclude, include, or modify the variable before shipping them to vendors. With this feature, you can prevent Google Analytics from accessing certain data or add more parameters (‘enrich’) to your events.

Transformation example 1: excluding country parameter

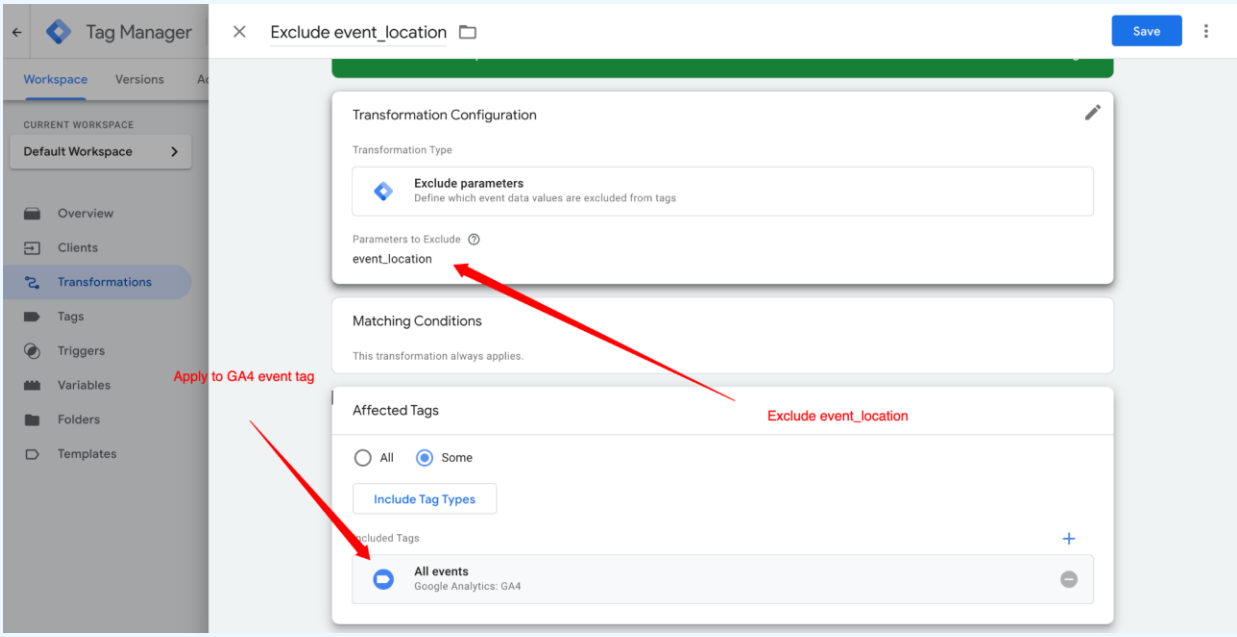
Let’s say we don’t want Google to access country level location. We can do this by heading to transformations section.



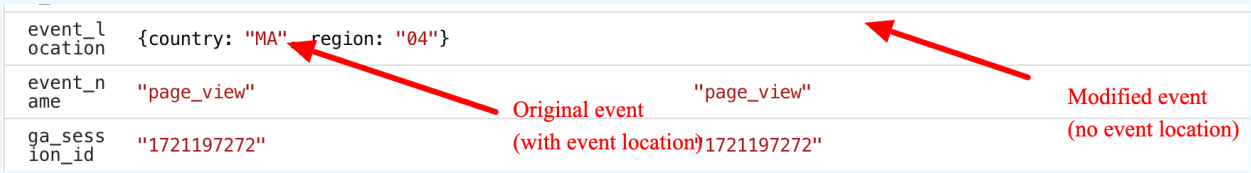
Next, we will add a new transformation:



We’ve got three options to choose from. Since we want to exclude a variable, we will select ’Exclude parameters’.

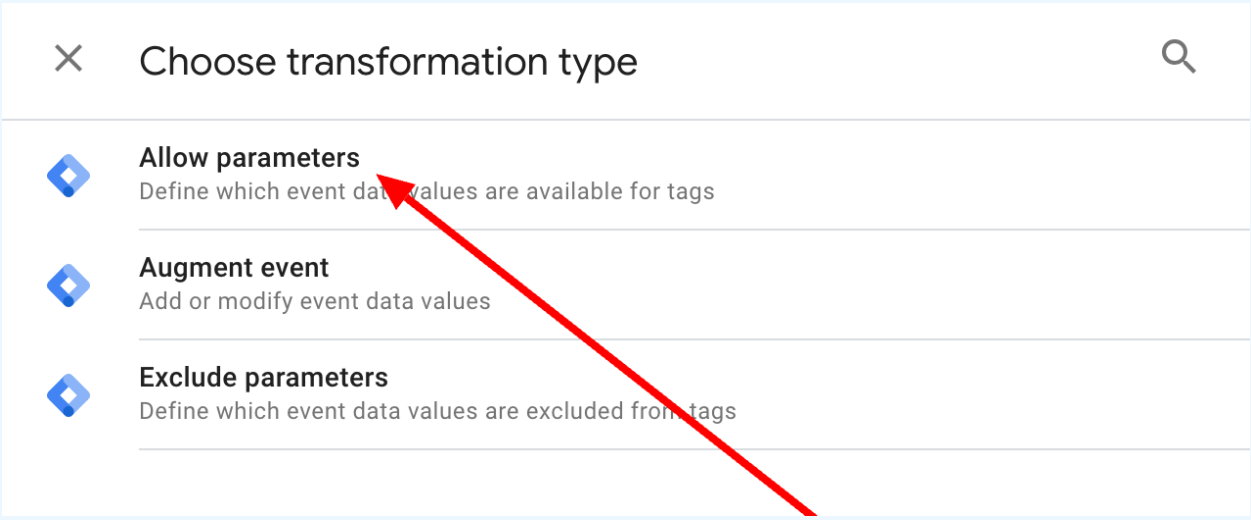


When you preview your server-side and open the tag where the transformation was applied, you will notice that the ‘event_location’ is excluded from the event we will send to GA4.

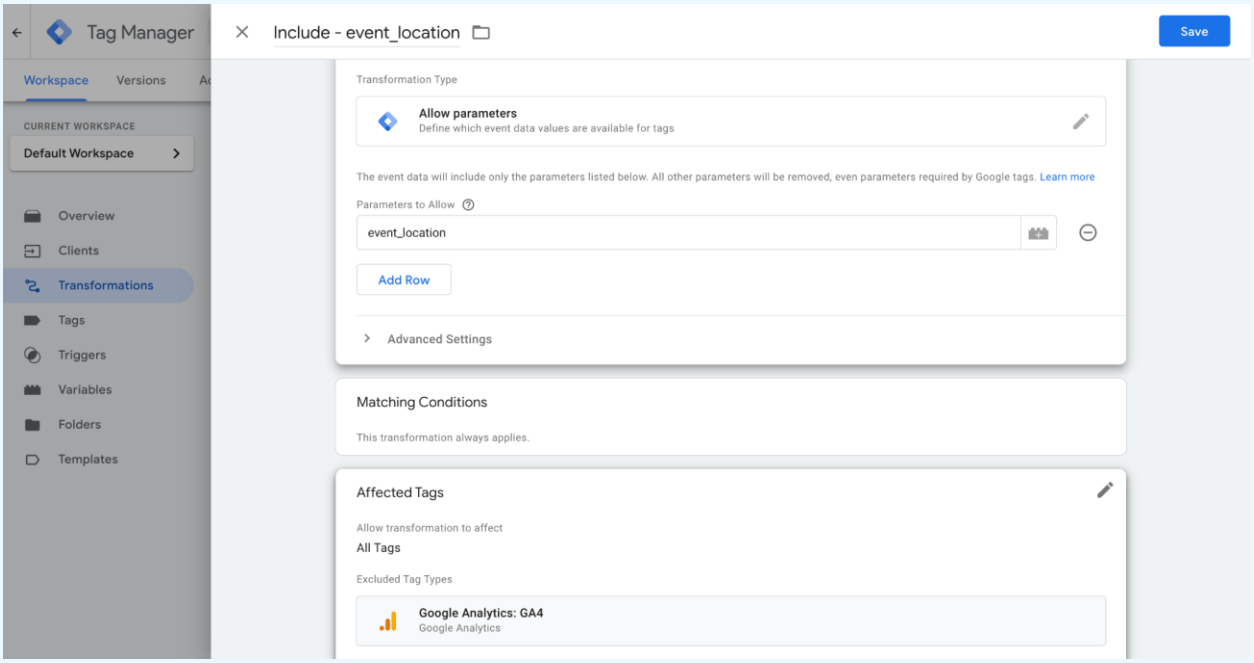


Transformation example 2: including country parameter

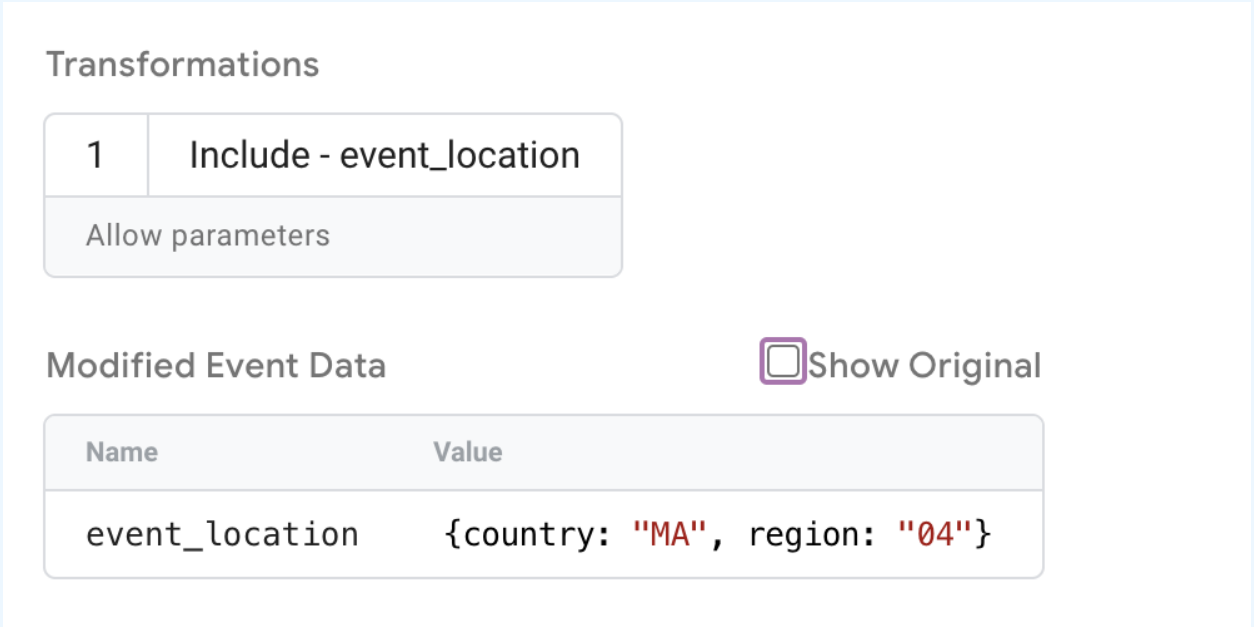
But what if we wanted to include only the event_location and ditch all the other parameters. Although this unlikely to be a real life situation, it will help you grasp the concept of allow parameters.



Next will add the event location.

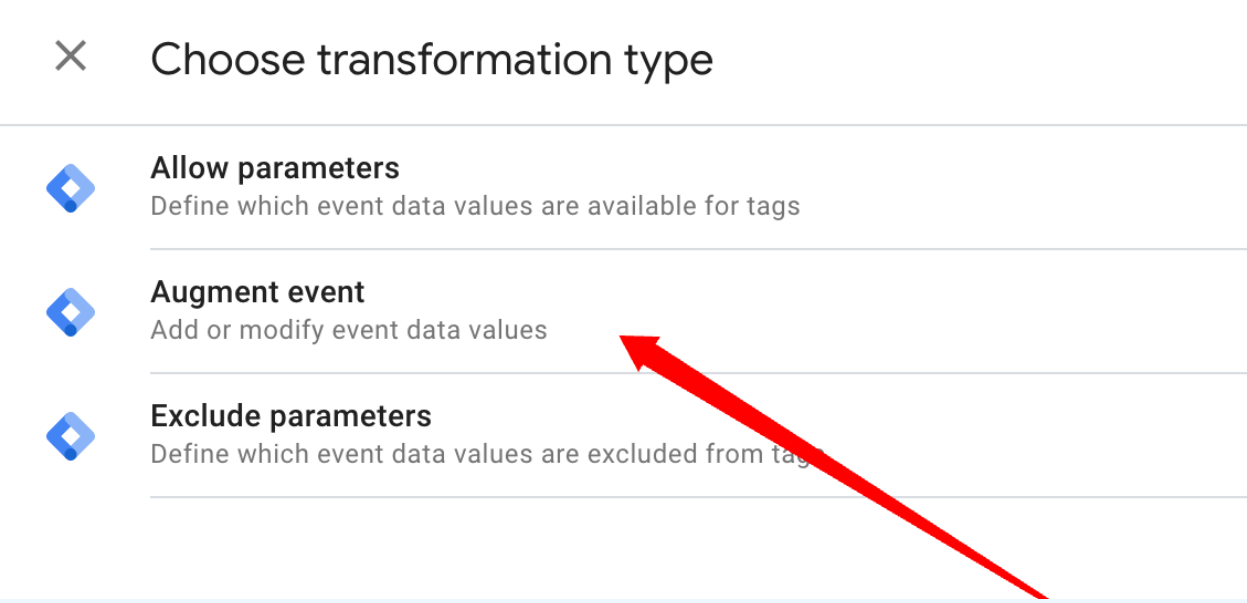


Let’s preview and see what happens:

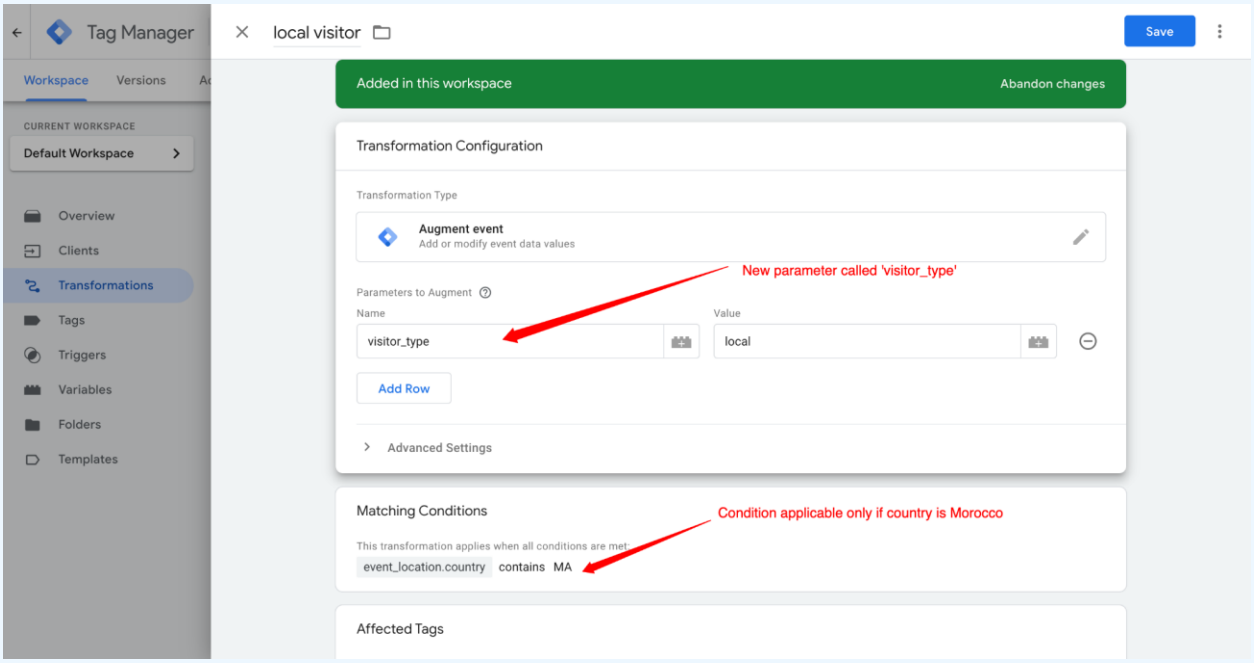


All other parameters are deleted from the new version of the event, except the ‘event_location’.

Transformation example 3: enriching events



Let’s say we wanted to label visitors incoming from our country as ‘local’. We can do this by adding a new augment event transformation.



- In the parameter to augment section

You add the field you want to change. In this case, I added a new field called visitor type. The user will be labeled as ‘local’ if the visit is occurring in my country.

- Matching conditions

The ‘Matching Condition’ is like the trigger section in the web GTM container. It will evaluate the user’s country origin. If there is a match, we will get the label ‘local’.

visitor_type	"local"
x-ga-are	"1"
x-ga-dma	"0"
x-ga-ecid	"2039453666"
x-ga-gcd	"13r3r3r3r5"

When I debug the set-up, you can see in the screenshot above that the new parameter was added.

Final words: where to go from here?

"Information is not knowledge. The only source of knowledge is experience. You need experience to gain wisdom."

Albert Einstein

My hope with this relatively short guide is not to teach you everything. I frankly get sometimes lost in what is supposed to be ‘my field’. What I tried to do through this guide is to go straight to the fundamental concepts that will allow you to dig deeper and learn as you go.

If you like to know more about server-side tracking and other digital analytics topics, I write frequently about these topics in my consultancy’s website. 99% of my content is public and free.

If you happen to be an ad/creative agency, and you are looking for a digital analytics partner to help your clients create and implement tailored Marctech/Adtech implementation, please apply here, or scan QR code below.

SCAN ME

Work with Hamza