Analytics mania

Server-side tagging with Google Tag Manager

and Google Cloud Run

Created by

Julius Fedorovicius

Founder of Analytics Mania

julius@analyticsmania.com

@fedorovicius

I can help you learn server-side GTM

Hi, my name is Julius. I founded Analytics Mania in late 2016. Since then, the blog has grown to hundreds of thousands of monthly visits. Here I share my knowledge, experiments, and learnings related to web analytics, mainly Google Tag Manager (GTM) & Google Analytics 4 (GA4).

I actively help others to learn Google Tag Manager with blog posts, Youtube videos and online courses. In my Intermediate/Advanced GTM Course, you will learn the following:

- How to configure server-side GTM
- How to optimize costs for hosting and handling data
- How to send data to multiple vendors server-side (e.g. Facebook CAPI, GA4, Google Ads)
- How to tackle more complex web tracking challenges
- Complete GA4 ecommerce setup and much more

You can **learn more about my Intermediate/Advanced GTM course here**.

Table of Contents

Chapter I

What is server-side tagging?

When Google launched the public beta of server-side Google Tag Manager (SGTM) in 2020, the audience split into two groups. Some people were cheering and claiming that this is a game changer while others were confused and asked "do we really need it?". This is understandable because server-side tagging is not easy.

To understand if SGTM is the right solution for you, first we need to start with the fundamentals. That's what this e-book is about.

However, remember that in this context, when I say "beginners", it refers to someone who does not know what server-side is. You will still need to have a good understanding of how website tracking with GTM works in general.

So, the ideal reader of this e-book should be someone who is pretty experienced with Google Tag Manager and wants to dive deeper into the world of SGTM.

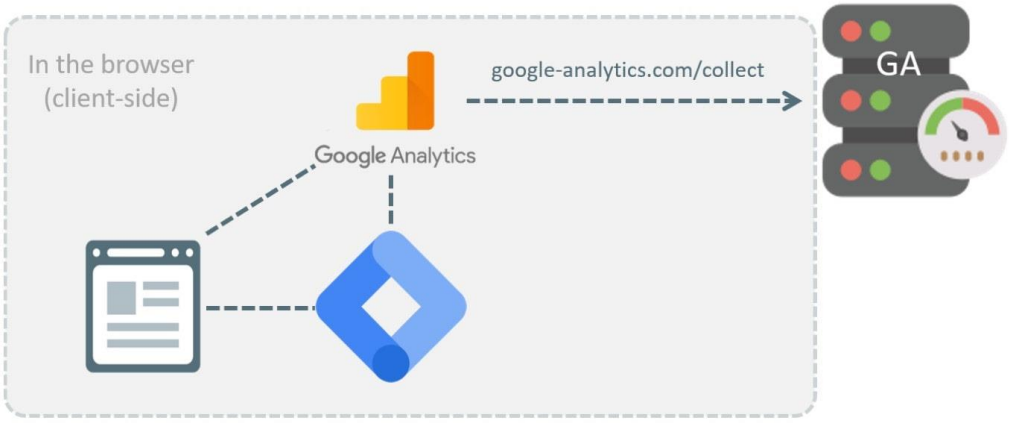
If your clients ask whether they need to adopt server-side tagging, you will be able to answer their questions after reading this e-book.

The classic way of Google Tag Manager

To explain how server-side tagging works, first, let's look at the setup/process that most of you are already familiar with deploying. If you want to install Google Tag Manager on a site, you need to add the web container's JavaScript snippet to the source code of your site.

When the page loads, the GTM snippet loads along with the rest of the webpage and fires tags based on your configuration. The Google Analytics 4 tag is one of them. All this activity happens client-side (read: *in the user's browser*). JavaScript codes load, collect data, and then send it somewhere else. For example, the GA4 snippet sends the data to *google-analytics.com* or *analytics.google.com.*

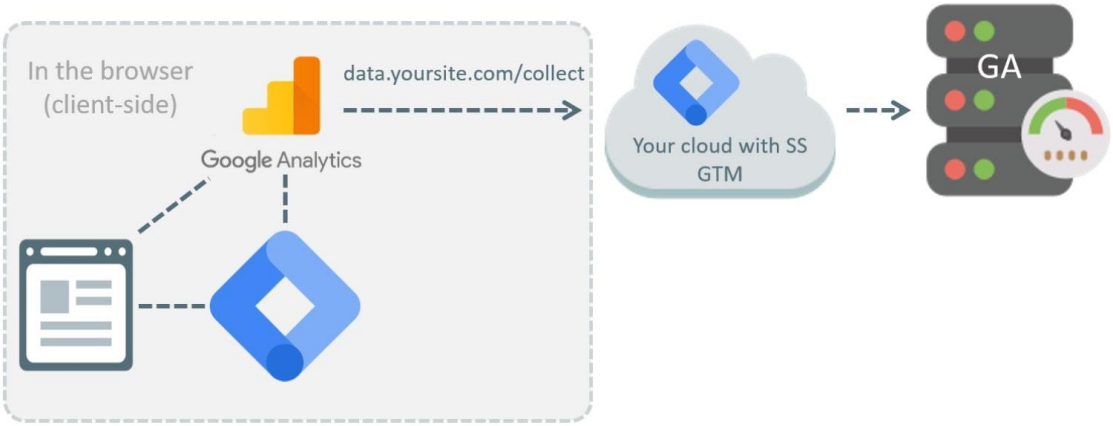
In addition to the data you instructed Google Analytics to collect (e.g., the click URL of the button), tracking codes usually gather additional information about the page, device, etc.



The server-side way

With server-side tagging, you are introducing an additional layer to the data collection chain - your own server (or servers) that hosts a server-side GTM container.

In that server, Google Tag Manager receives data from the client-side tracking codes, processes it, manipulates it (if needed), and then sends it to the 3rd party vendors or maybe even in-house analytics tools you are using.



Therefore, we have client-side tracking, but we also have some server-side parts. And if you want to go even further, you can skip the client-side tracking and get data from one server to your GTM server-side setup and send it to some marketing/ analytics end-points. But in this guide, we are focusing on the mix of client-side and server-side tagging.

Why would you need this?

You are probably thinking now: why would you want to add an additional step to the data collection process? Why should you bother having a server(s) in the cloud where another GTM container is loaded that sends the data to other vendors?

This is a very legit question. Let's dive right into the benefits.

Benefits of server-side tagging

There are many benefits you can get from server-side tagging. I will mention only the most important ones (in my opinion).

Reduced load on a page

When a web page loads, it has to download and process a lot of resources. As my experiments proved, the JavaScript snippets you add for analytics/marketing purposes can significantly slow down your site, especially if you add new scripts more often than Buckethead releases new albums).

Even if the script is loaded asynchronously (which is pretty much the standard today), it can slow down your site.

Imagine this: instead of loading dozens of different scripts on your website, you could have just one. That script would send data to your server-side GTM container.

From there, the container would modify the data according to your configurations and then send it further to more tools/platforms you are using (analytics, marketing, CRM, you name it).

However, configuring a server-side container adds another level of complexity because you will need to configure how to transform the data, etc.

At the moment, that is not an easy task — you will need to learn how to code JavaScript, understand how server-side tagging works from a technical perspective, be familiar with cloud platforms, etc.

But the community keeps creating new plug-and-play templates that will do the hard work for you, and you will need to click some buttons/checkboxes, etc., and launch without too much hassle.

Control what data is sent to vendors

As I mentioned in one of the previous chapters of this e-book, JavaScript libraries loaded on a site will likely track more than what you instruct them to do. These libraries also collect details related to the device (for example, screen size), page details (page URL, page title), etc.

Libraries related to advertising might collect everything they can to identify visitors with higher precision. Data collection involves various techniques, including fingerprinting.

From a privacy standpoint, you would want something other than that, as it increases the risk of leaks of personally identifiable information (PII). With server-side tracking, you control what reaches those vendors.

Here's an example:

You have some custom-built JavaScript library that is collecting data about visitor behavior on your site

That library sends the data to your server-side Google Tag Manager container

You configure that container to send the received data to:

- Google Analytics
- Facebook Pixel
- Google Ads

In this case, these 3 analytics/advertising platforms can only get additional data about the visitor if you send it to them. If you send the user ID, then that ID is delivered. However, optional data that standard JavaScript libraries could have accessed on a website is now out of reach.

Why? Because Facebook Pixel in this setup is communicating with your cloud server. It cannot capture data from the visitor's browser.

If your custom JS library on the site collects some PII, you can process the requests in the server-side container and remove/hash them.

Reduce the impact of ad blockers

Here's a situation: you got consent from a visitor for analytics tracking, but that visitor is using some adblocking extension. The extension blocks all the requests to Google Analytics, as it blocks ads.

Because the request was blocked, the browser will not send any data collected by the various tags on your website to Google Analytics.

With server-side tagging, you could create a custom subdomain to which you can send the data, e.g., *analytics.yourdomain.com*. A GTM server-side container is waiting for the data at the other end. Once it is received and processed, the container sends the data further to Google Analytics.

Currently, most ad blockers do not block requests sent to domains like *analytics.yourdomain.com*. Who knows what will happen in the future, but this solution works now.

Extend the cookie expiration on Safari (ITP)

Intelligent Tracking Prevention (ITP) is a collection of privacy-enhancing features in the Apple Safari browser (but it also affects more browsers on iOS). Unfortunately, many things are affected/limited to prevent cross-site tracking, but my concern for this e-book is cookie expiration.

If the 1st party cookie (meaning that it can only be accessed on your site) is stored in a visitor's browser by JavaScript, that cookie will expire in 7 days (in some cases, the limit is 24 hours).

Even if that JavaScript is entirely custom, built in-house, and is not related to any 3rd party vendor, that cookie will expire in 7 days (unless a visitor keeps revisiting your site and that cookie gets updated).

For the time being, you can avoid this 7-day limit if you store the cookie on the server side. In other words, if a server (accessed via your domain) stores a cookie in a visitor's browser, its expiry date is whatever you configured it to be (e.g., 2 years).

I explain how to extend cookies with SGTM in my Intermediate/Advanced Google Tag Manager course.

Update: *Apple launched an update to limit the duration of cookies set from the server. If the website server's and your SGTM server's IP address's first two numbers don't match, the cookie extension will not work.*

If the website server's IP address is 1.2.3.4 and SGTM server's IP address is 44.55.6.7, the first two numbers (1.2 vs 44.55) do not match, thus, cookies set from the server will still expire in 7 days.

A solution for that would using the same load balancer for your website and for your server-side Google Tag Manager. Then, IP addresses will match and the lifetime of cookies (set from the server) will be extended properly.

Drawbacks

However, not everything is perfect here. There are several drawbacks as well. Or maybe they should be called "Things to keep in mind".

In fact, I already have a separate section for "things to keep in mind," and it contains more notes. Keep reading.

Paid solution

Don't get me wrong. GTM server-side container is free, and you will not incur much cost in most cases in the testing configuration.

But if you want to run it on a live website that is getting decent traffic, you will need to open your wallet.

No, that's not because the GTM container requires some premium GTM subscription. However, you need to host that server-side container somewhere. And that's where you spend the real $$$ (on Google Cloud Platform, Stape, Azure, AWS, or somewhere else). Even though Stape is cheaper than the other options I mentioned, it will still incur costs.

The price depends on a lot of factors:

- Which hosting provider do you choose
- The number of requests or traffic you send/get
- Number of servers, etc.

I am most familiar with Google Cloud Platform (GCP) and Stape. The minimum recommended configuration on GCP would cost you at least $90/month. On Stape, the pricing starts at $20/month.

But if you get more traffic, your expenses will be higher.

You can also use your servers (by doing a manual setup). But that will also cost you something.

Not for everyone

To start working with server-side tagging in Google Tag Manager, you will need to become even more technical (or hire someone experienced in analytics development).

If you thought that GTM already requires a lot of technical topics (spoiler alert: it does), then from now on, the rabbit hole goes deeper.

Chapter II

How to configure server-side GTM

How to configure server-side tagging in Google Tag Manager?

Now, let's dive into your first setup. I'll try to keep things as simple as possible (even though, in reality, they are more difficult).

Here's what the process looks like in a nutshell:

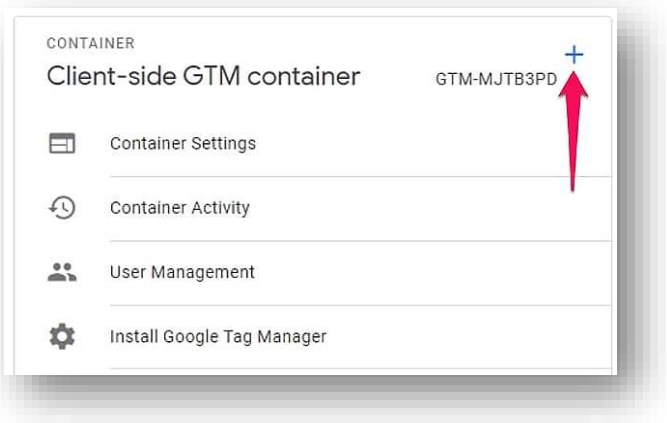
- Create a server-side Google Tag Manager container + cloud project
- Configure tags, triggers, variables (if needed), clients (I'll talk about them a bit later) in the server-side container
- Send the data to your GTM server-side container
- Test and check

By default, server-side Google Tag Manager uses Cloud Run (it's a part of the Google Cloud Platform), and that's what I will be using in this article, too.

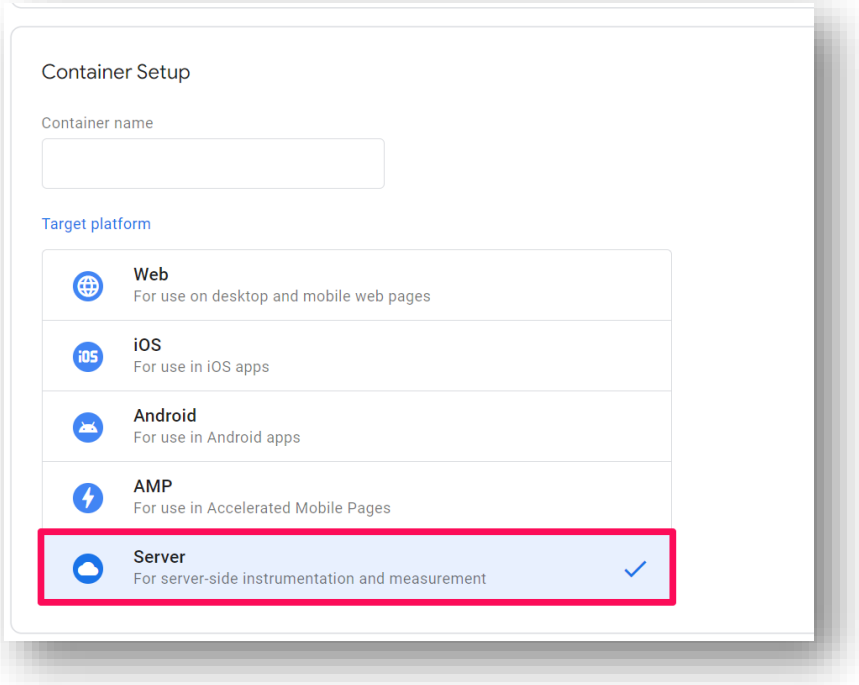
Note: The process of setting up the server differs with each hosting provider. So, if you decide to go with, say, Amazon AWS, you'll need to google the instructions for that.

Create a server-side Google Tag Manager container

Login to your Google Tag Manager account, go to Admin and click the Plus icon.

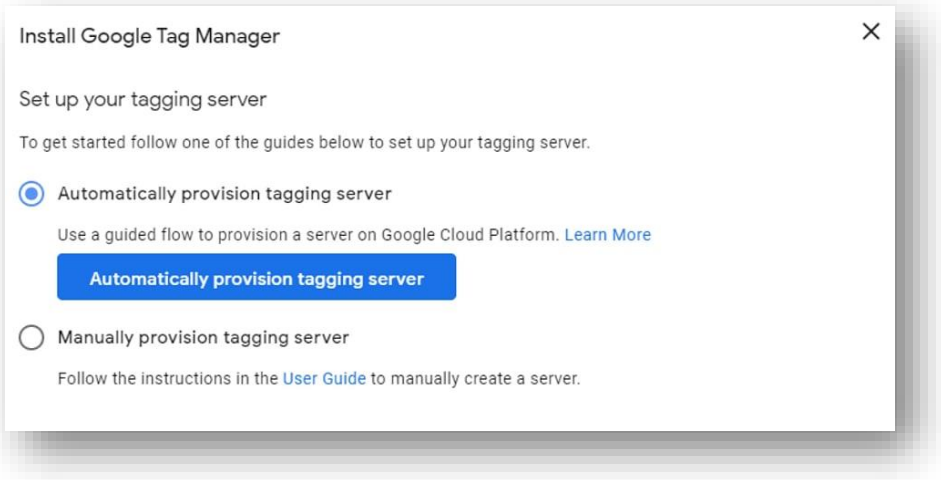


Then enter the name (for example, Demo Server-Side Container) and choose the type “Server”.

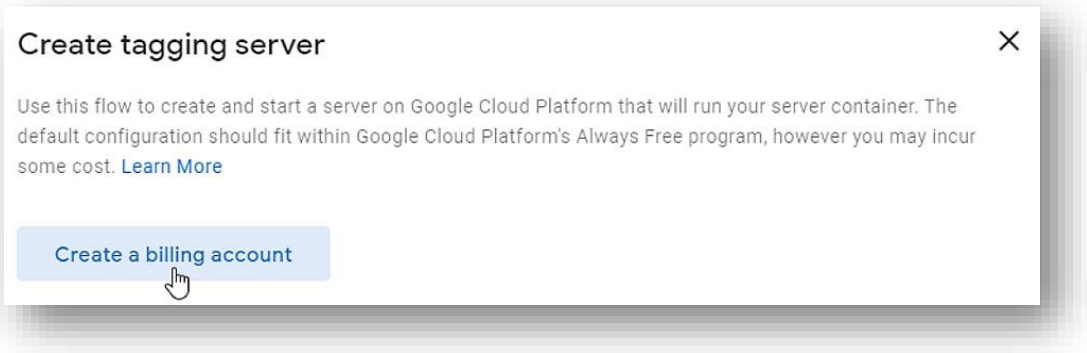


Then, you will need to create a new project in the Google Cloud Platform (because you need to host the new container somewhere). Luckily, the process here is pretty straightforward. You will need to enter certain information and follow all the steps to finish the task.

First, the window will ask if you want to automatically get a new server for the container or manually create it. The first option involves a much simpler process; choose that (but if you plan to use, say, Stape, you will need to use the manual provision. But, as I have said, look for instructions of that specific provider).



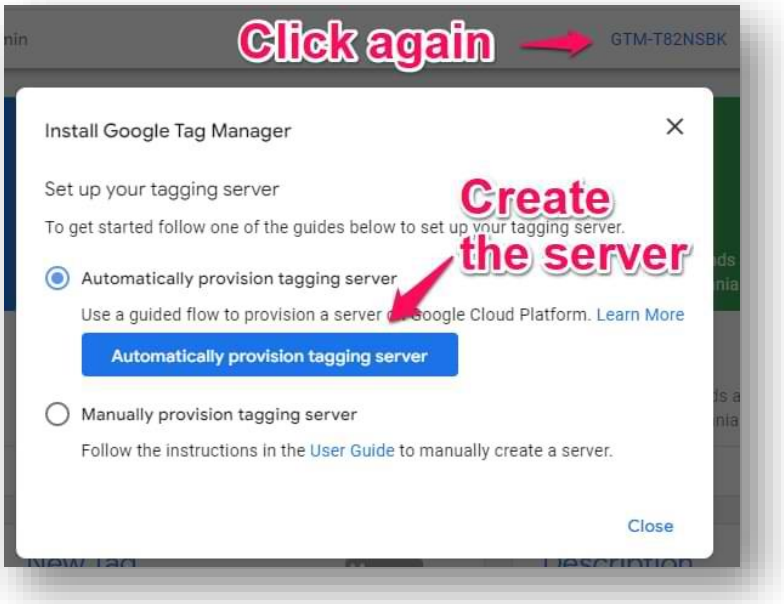
If you are new to the Google Cloud Platform, you'll need to create a billing account and enter your credit card details. When you see the option "Create a billing" account", click on it.



And then, follow all the steps the platform asks you to complete. Once your account is ready, you will be redirected back to GTM.

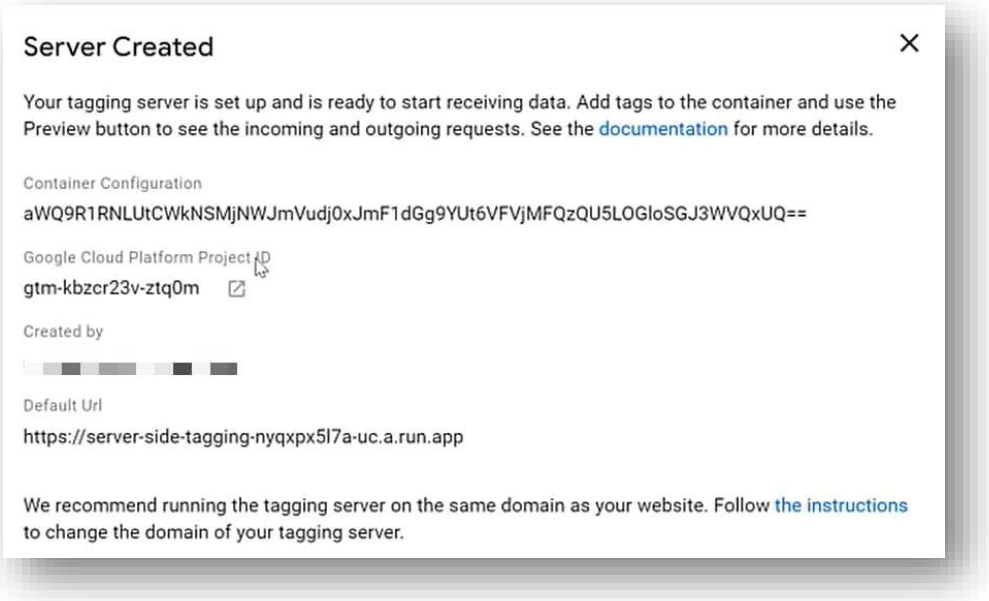
It should automatically resume the server creation process. But if you are back to GTM (but don't see anything related to the new server), open your new SGTM

container, click the Container ID, and then Automatically provision the server once again.



The loading process might take several minutes, be patient.

Once complete, you will see the information about the created server (including the Google Cloud Platform Project ID, default URL, etc.)



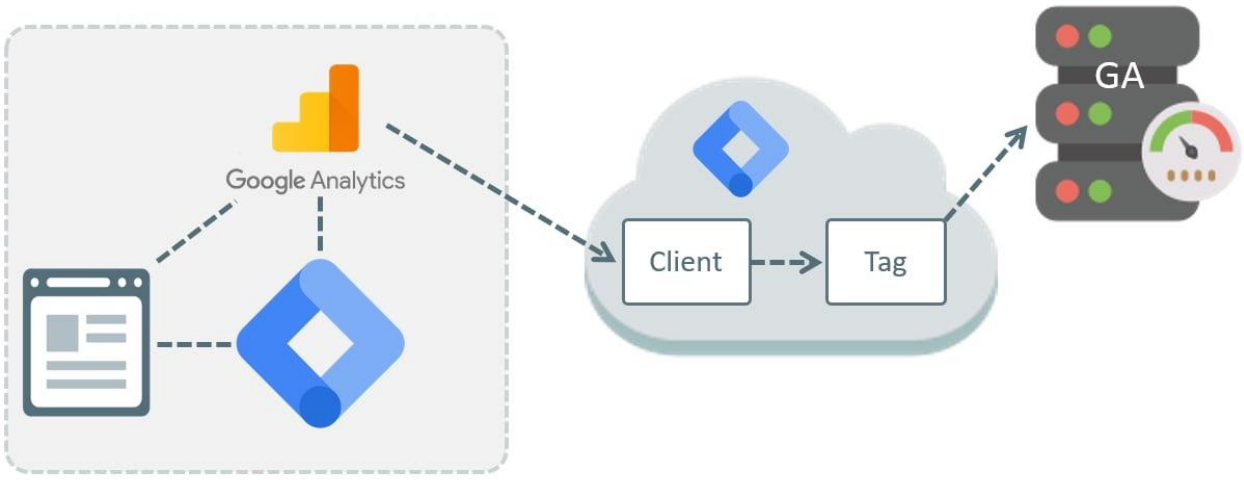
Configure tags, triggers, variables (if needed), clients in the server container

The concept of tags, triggers, and variables is also available in the server-side containers too, but there is one new thing that you need to become familiar with.

Client.

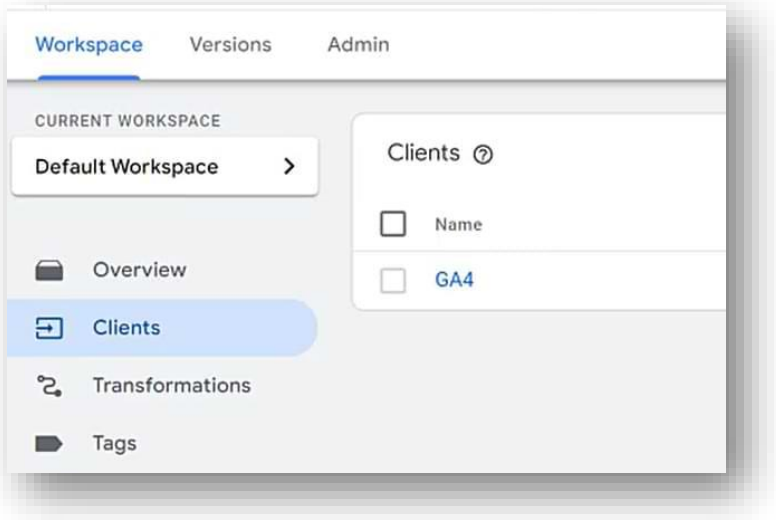
No, this is not a customer.

Clients are responsible for getting and processing/adapting the data received by the GTM server-side container. Then it makes the data available to tags, triggers, and variables in that container.



For example, the data is usually available in the regular web container because some code pushes it to the data layer. In this case, the server-side container receives the data from *somewhere*. The client then catches, adapts, and makes it available in the container.

When you create a new server container, it creates one client automatically. It's related to GA4.



You can additionally create more clients if you need to. Still, the number of client templates available is limited now (however, in the future, we should expect more because the community should build something).

Also, you can create your own clients, but this requires good JavaScript knowledge + becoming familiar with the APIs available in the Client Templates.

We don't need to create or code clients for your first setup. Instead, we will be using the existing Google Analytics 4 client. Every time GA4 sends data to the GTM server container, this client will catch it and make it available for use to tags, variables, etc.

Note: if you want to learn much more about server-side tagging, you should check out my Intermediate/Advanced GTM Course.

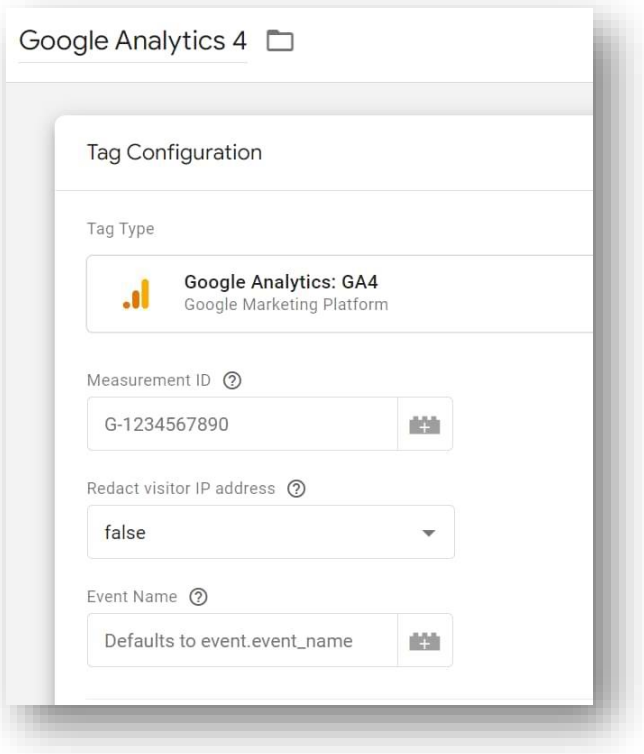
When that happens, we need to activate a Google Analytics 4 server tag to send the data further to Google Analytics servers.

To do that, go to *Tags* > *New* > *Google Analytics 4*. This tag will forward all the data from a Google Analytics 4 client.

In the regular web GTM Container, you are accustomed to creating different tags for different interactions:

- One tag for pageviews
- One for purchases
- One for outbound link clicks, etc.

In the server-side case, it's enough to have one GA4 tag (at least in the early stages of your SGTM setup). This tag will receive data about GA4 events, pageviews, purchases (from your website), etc., and will forward them to Google Analytics servers.

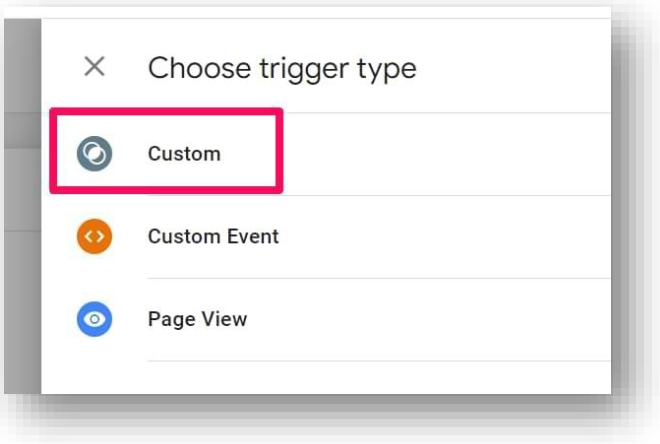


When does it make sense to create multiple GA4 tags in the server container? Sometimes, certain Google Analytics 4 events may require you to override settings. Then, it makes sense to create a separate tag for X events.

But in this first setup, we don't need to make any tag configuration changes. Instead, let's move to the trigger by navigating to the "Triggering" section.

You will see just three pre-built triggers, just like with the client-side setup - *Custom, Custom Event,* and *Page View*. For our particular case, we need to create a trigger

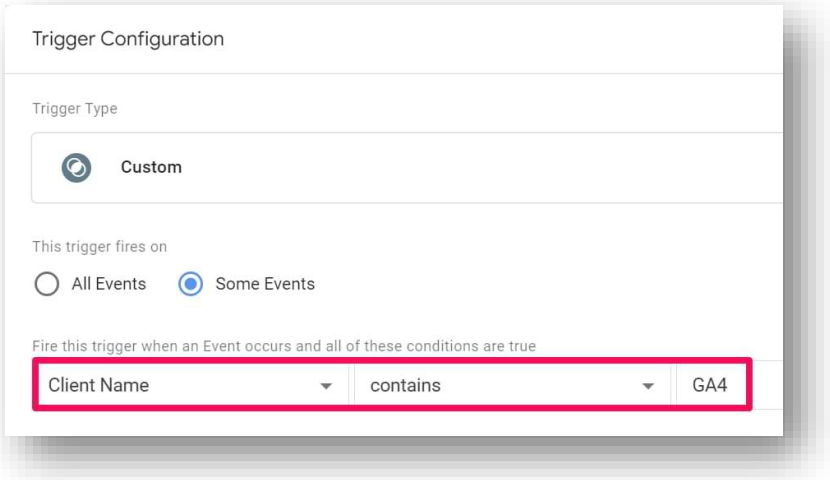
when GA4 appears as the Client name for the associated event. Here, what you need to do is select the *Custom trigger*.



Custom means "A request was sent to the GTM server container". If you're not concerned about which requests are sent to Google Analytics, you can keep using "All events", but it would be better if you were more specific.

What if we have multiple clients in use in the future (which is very likely)? If this is your case, you could limit this trigger to activate only when the default Google Analytics 4 client catches the request.

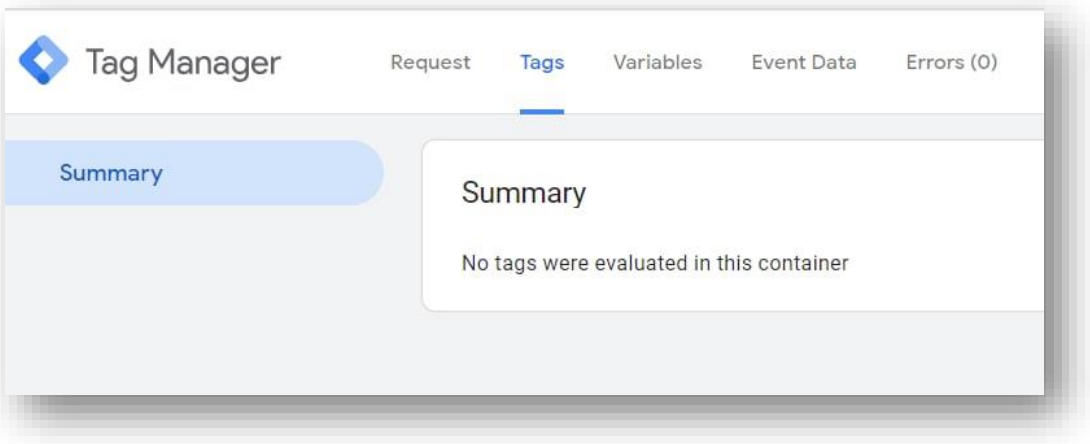
That's why you should enter the following condition: **Client Name contains** (or exactly matches) **GA4**.



P.S. *Client Name* is a built-in variable you must enable in the **Variables** section of the server container.

Save all these changes. It's time to enable Preview and Debug mode. The preview mode will open in a new tab and looks quite similar to the one you are used to seeing in the Web container. However, there are several new things/differences. I'll mention them a little later.

At the moment, you won't see any data there (because nothing has been sent to this container yet).



But that's about to change soon.

Send data to the server-side Google Tag Manager container

There are several ways how to send data to the GTM server-side container:

- A developer can edit gtag.js code snippets added to your site's source code
- A developer can write some custom code/library that prepares and sends the data to the server container itself

- You can configure a Google Analytics 4 tag in Google Tag Manager's web container.

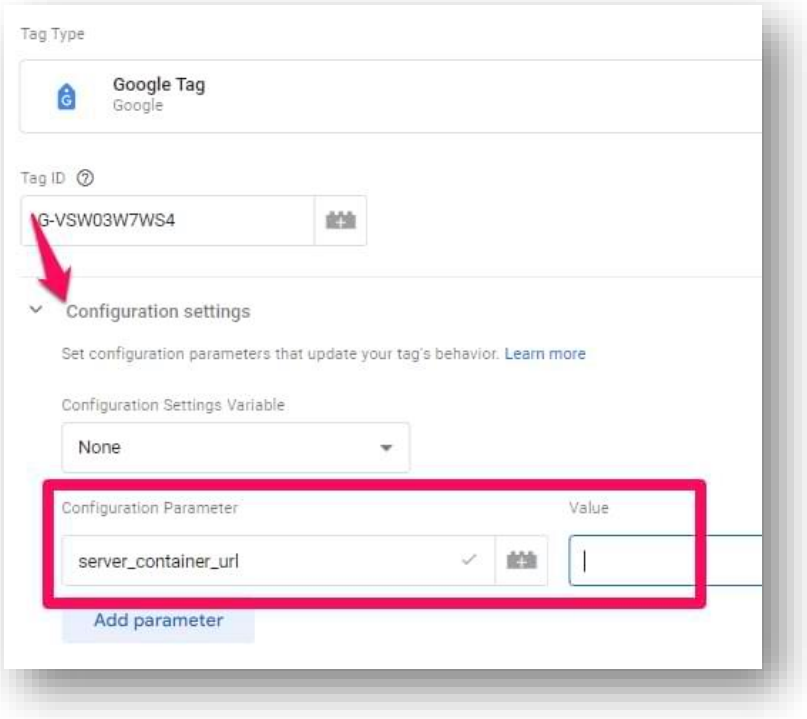
To no one's surprise, I'll use the third option right now.

Let's say you have a website where a regular GTM web container has been added. You are already firing a GA4 tag on every pageview. Most likely, you also have some additional event tags.

All of them are sending data (by default) to *google-analytics.com/collect* (or */r/collect,* or *analytics.google.com/collect*, etc.). Now, we should change that destination URL and forward all requests to your fresh new server-side container.

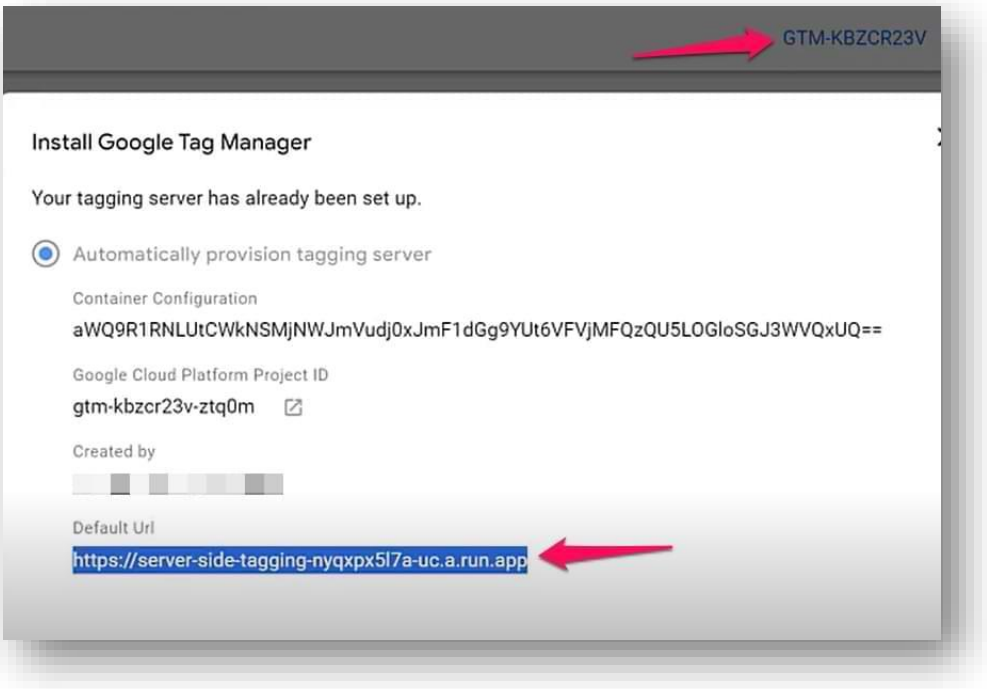
How do you do this?

First, we need to edit the Google Tag (which is used to install GA4 with GTM). In your web container, open the Google Tag that is responsible for activating Google Analytics 4. Then click *Configuration settings > Add parameter* and enter server_container_url.



That's where we will need to enter the URL.

You will likely want to send ALL GA4 requests to the server-side container. What URL should you enter? Let's go to the server container and click the container ID:



In the popup, you will see the Default URL, copy it, and paste it to the Server Container URL in your web container's Google Tag.

In the future, you should do the same thing with all GA4 event tags. There, you should insert the **server_container_url** parameter in the *Event parameters* section.

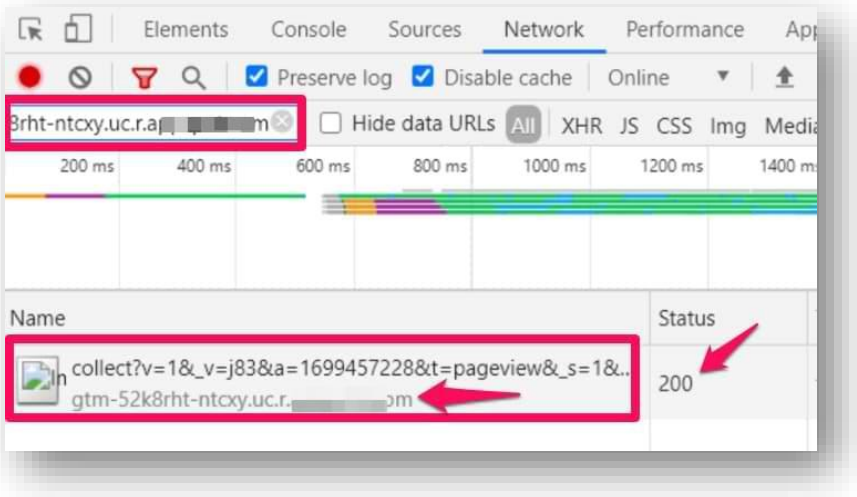
Time to test

Refresh the preview mode in your server-side container (by clicking the "Preview" button in the GTM interface once again).

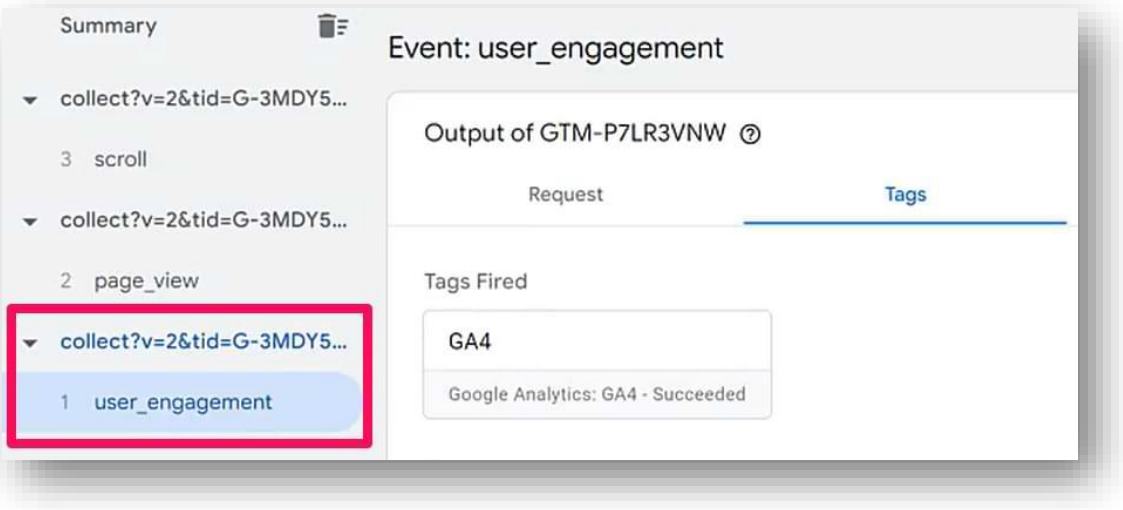
Enable the Preview mode in the web GTM container.

Firstly, a Google tag in your web container will fire (if you have set it to fire on All pages).

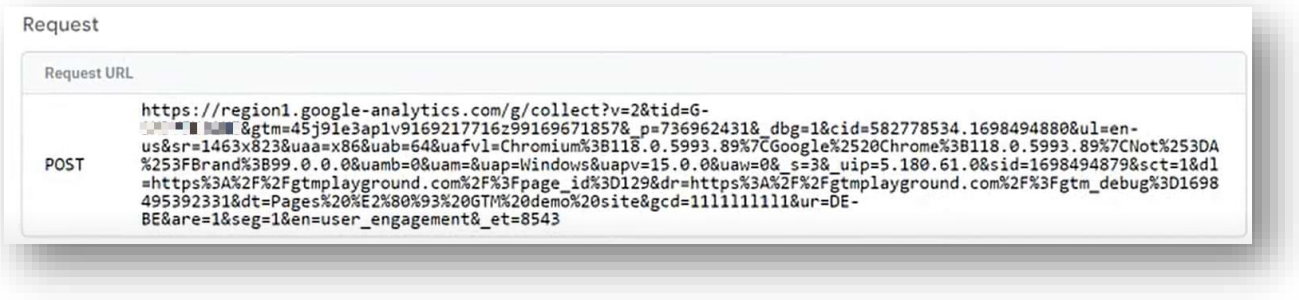
You can also check the Network tab of your browser developer tools. Enter "*/collect*" (without quotation marks) to find the request. For example, suppose you can't refresh the page once again (while */collect* is still in the search bar). You should see the request sent NOT to google-analytics.com but to your container's domain (should contain *[…]run.app*). If the status of this request is 200 or 204, that's good.



Then, you need to go to the preview mode of the server GTM container and see if the request was received. If yes, it will display on the left side of the preview mode.



Click it and check whether the GA4 tag in your server container fired. You can click the tag and see what kind of data was sent further to Google Analytics.



Also, feel free to explore other tabs of the preview mode, e.g., Event Data. Think of this tab as a cousin to a Data Layer tab in the web container's preview and debug mode.

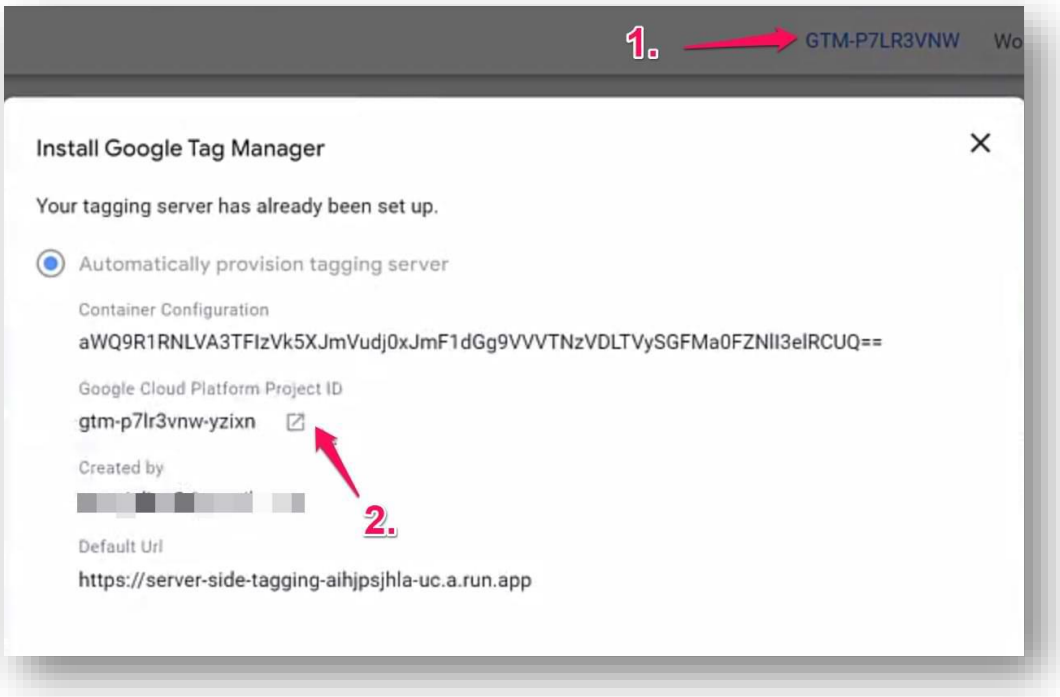
Finally, go to your GA4 real-time reports and check whether the data is visible.

Connect your custom domain

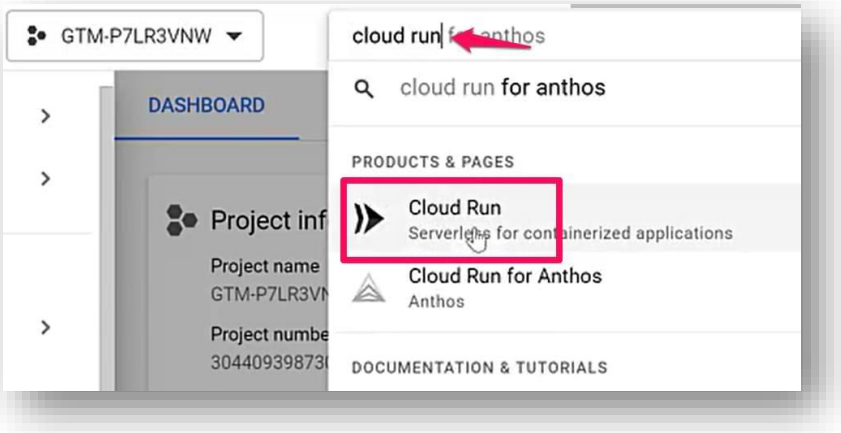
The request to the GTM server container we sent now was thanks to a 3rd party domain, which belongs to Google (*[.....].run.app*). Hence, this is still considered as a 3rd party tracking. But because we already know that browsers are constantly tightening the rules around web tracking, I you absolutely must use your custom domain (as the endpoint) instead.

In other words, you should send requests not to [.....].run.app but to your subdomain, e.g., *analytics.yourdomain.com*. That way, website measurement will be happening in the first-party context.

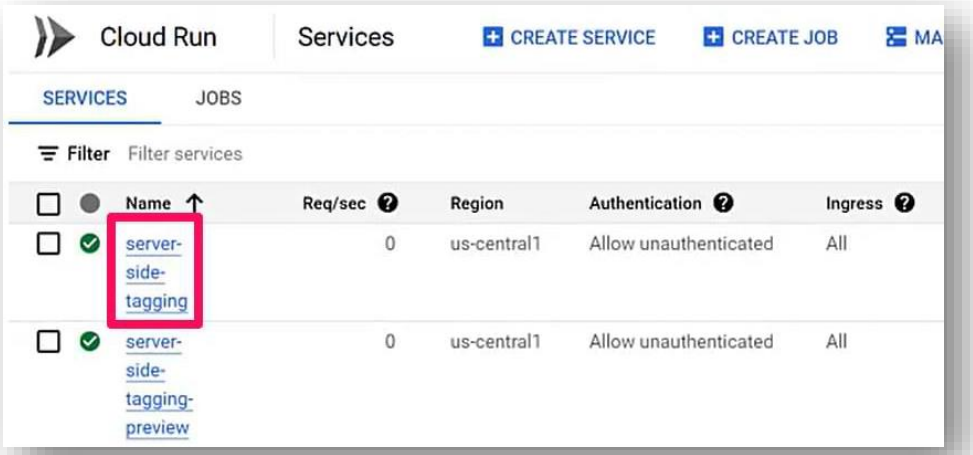
The mapping of a custom domain will differ depending on which hosting provider/solution are you using. In the case of Google Cloud Run, the process goes like this. Go to your server Google Tag Manager container, click the container ID and then click the Google Cloud Platform project ID.



You will be redirected to the Google Cloud Platform. Use the search bar at the top of the interface and look for *Cloud Run*.

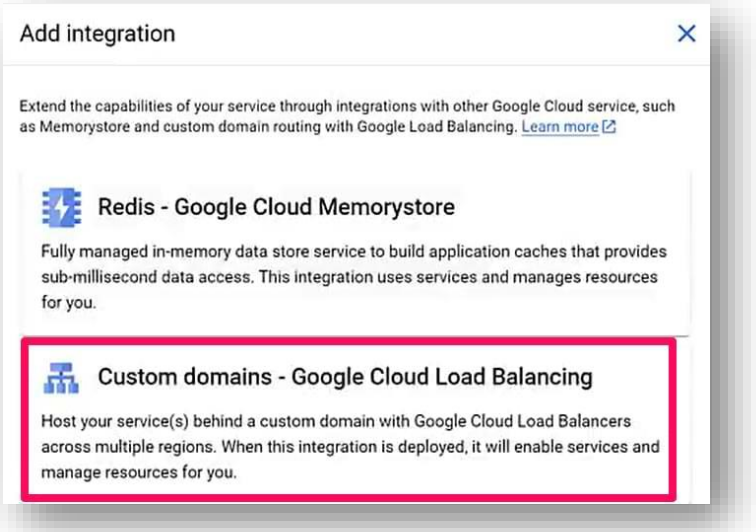


Here, you will see two servers. One is for the preview mode, and the other one is for the live environment. Select the tagging server (not the preview server).



Then click Integrations and Add integration.

Select *Custom Domains*.

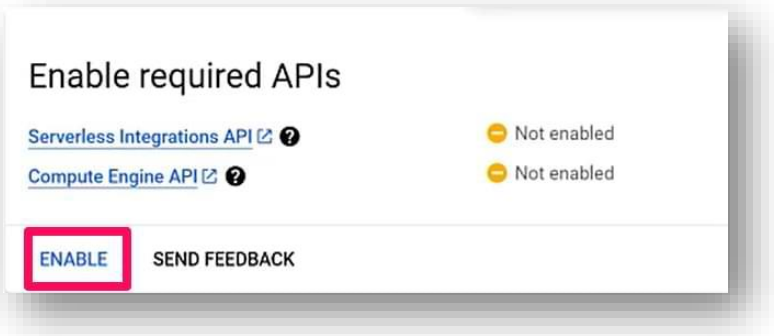


Enter the subdomain that you are planning to use for this server. For example, if a website can be accessed via *www.example.com*, you can use a subdomain *analytics.example.com* or *measure.example.com* (or something else).

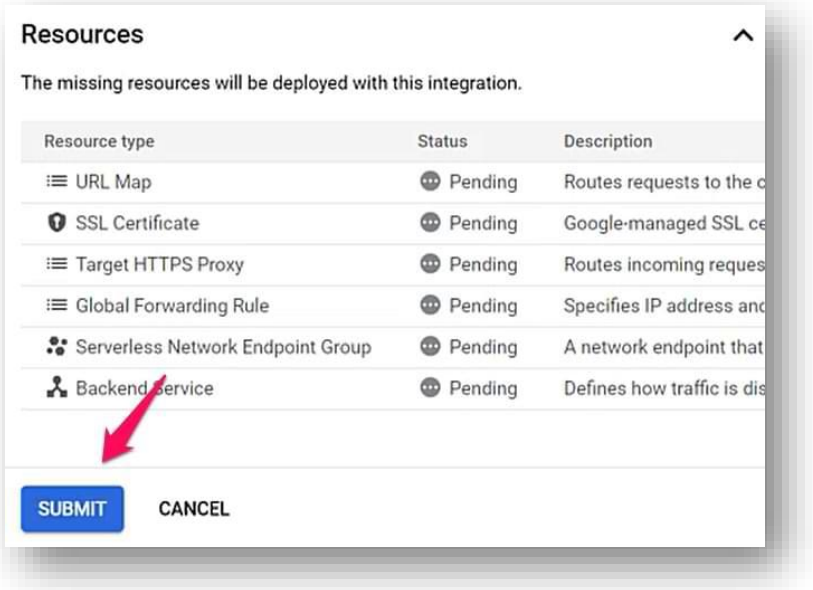
Alternatively, you could use the same domain as your website (and then have a dedicated path, e.g., *example.com/analytics*). This is called *same origin* and it requires more configuration/involvement from your developers. So, let's use the subdomain option now.

In the Service column, make sure that you have selected the live tagging server (not the preview server).

If you see a notification that some APIs must be enabled, do so. Click *Enable*.

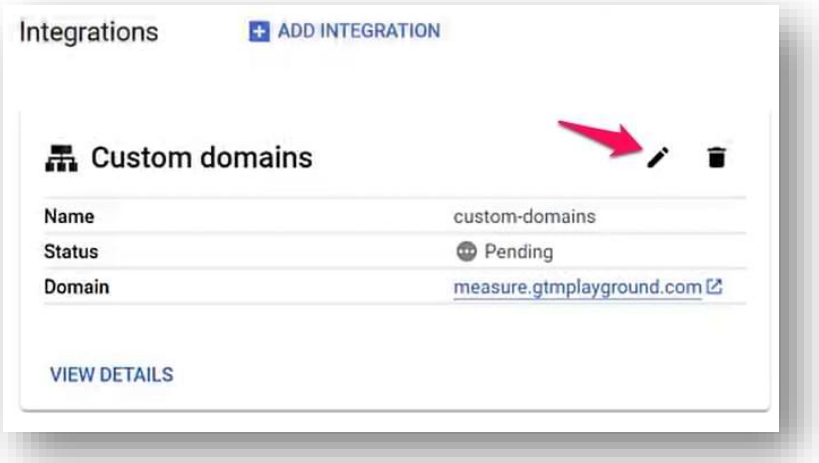


It might take several moments for this to complete. Then click *Submit*.

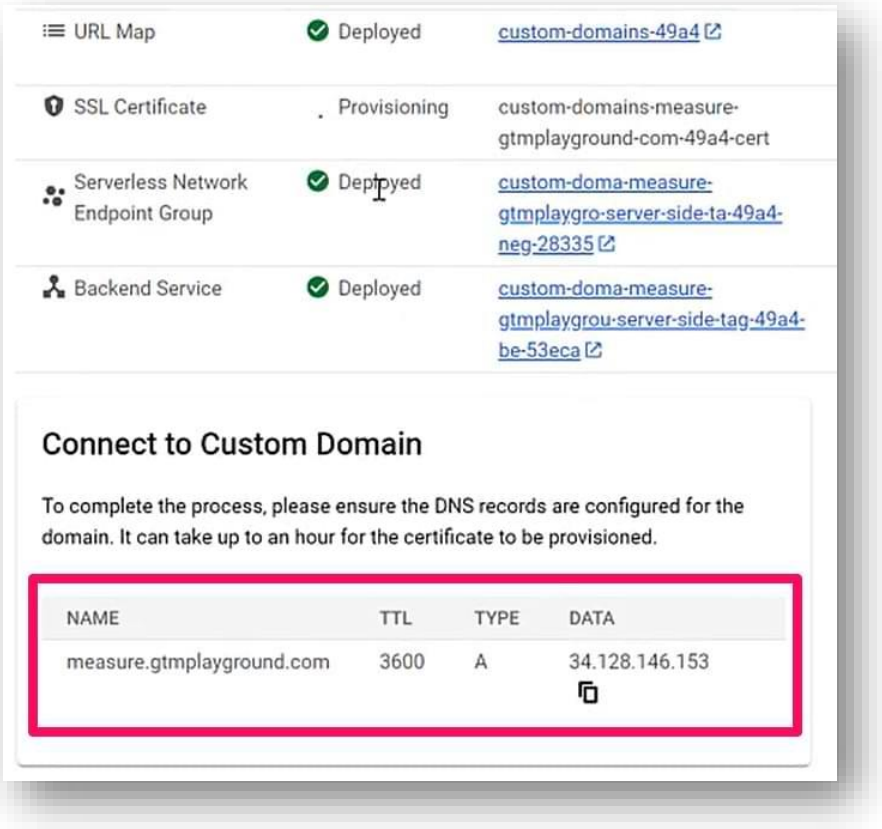


It might take several additional moments for this configuration to be created.

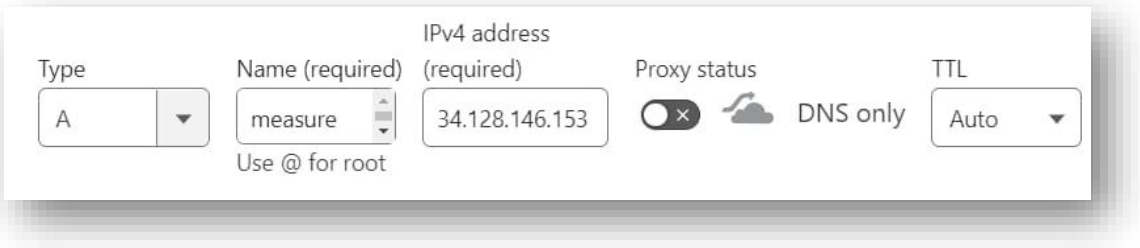
We're halfway there. After that, click the Pencil icon in the list of your integrations.



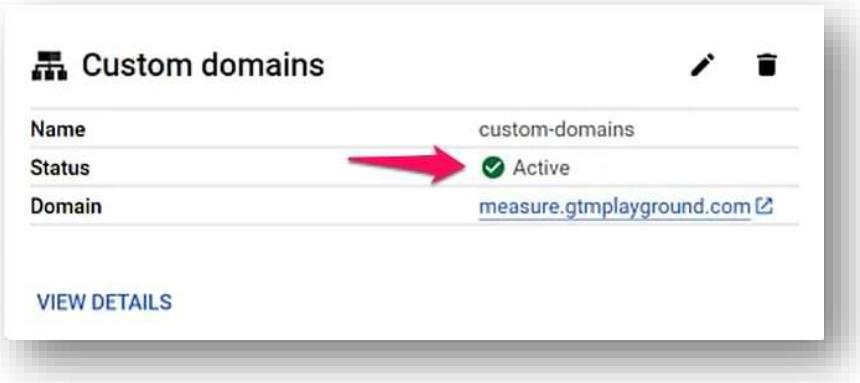
This will show you that most of the recourses are complete (except one - SSL certificate). This will become ready when you update your DNS records). At the bottom of the custom domain settings, you will find the instructions.



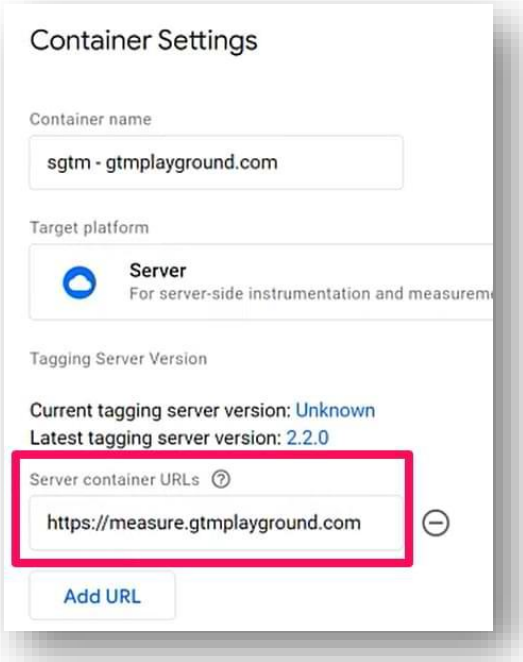
You (or your developer) must create an A record in your domain's DNS records. The process here differs depending on where you manage your domains. I am using Cloudflare, so the settings would look like this:



Save the changes and wait until the SSL certificate resource is also processed by Cloud Run and the integration is active.



We're almost done. Now, go to your server-side Google Tag Manager container > *Admin* > *Container Settings*, and in the *Container Settings URL*, enter your custom subdomain, e.g., https://measure.example.com (without a trailing slash).



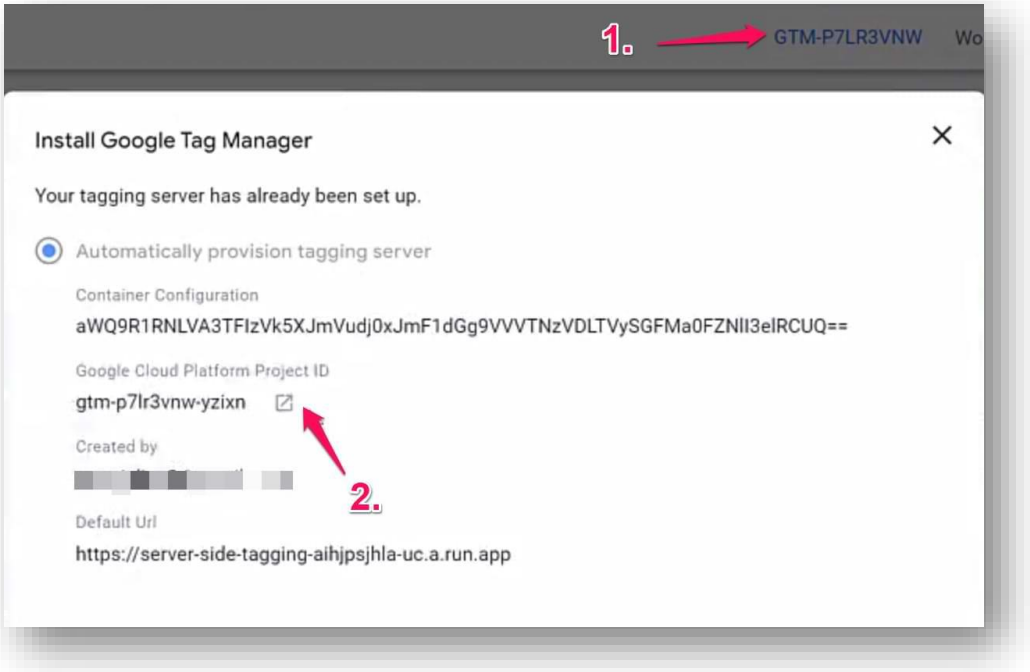
Finally, go to your website Google Tag Manager container and open Google Analytics 4 tags. In the *server_container_url* field, enter your custom subdomain (e.g., https://measure.example.com).

If you have more tags, for example, GA4 event tags (and you want them to send data to SGTM), all of them should be updated to use your custom subdomain. **Note:** not all vendors support server-side tagging, so some tags cannot be updated for this.

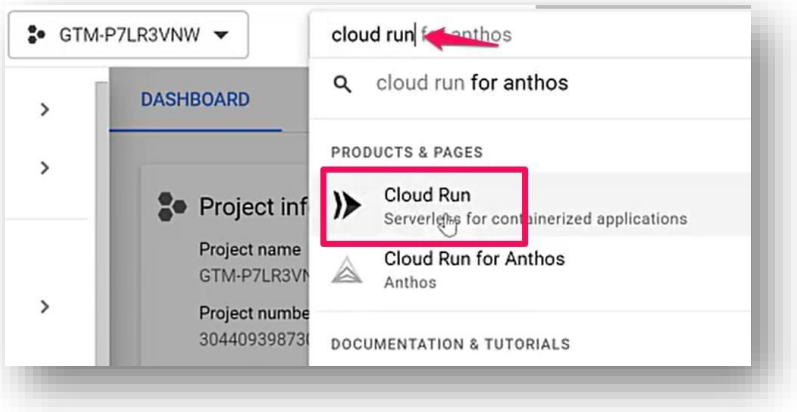
Upgrading the server

One of the final steps in this guide is to upgrade the servers (so that your SGTM can support the website's traffic and have enough power to process incoming requests).

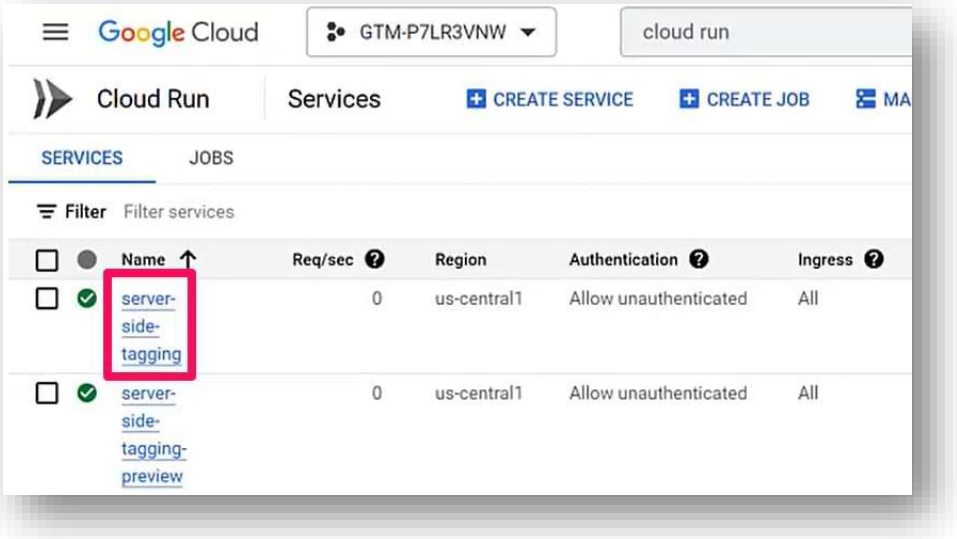
Go to your server Google Tag Manager container, click the container ID and then click the Google Cloud Platform project ID.



You will be redirected to the Google Cloud Platform. Use the search bar at the top of the interface and look for *Cloud Run*.



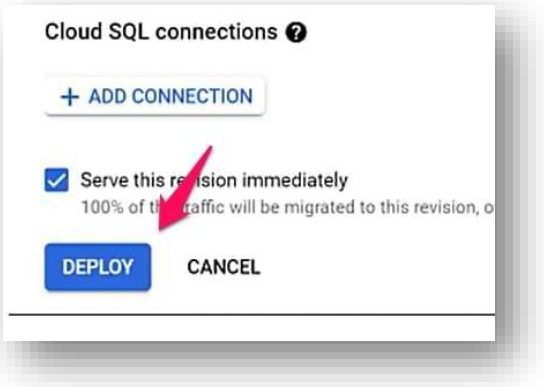
Select the tagging server (not the preview server).



Then select Edit & Deploy new revision.

In the *Autoscaling* section, change the minimum number of instances to 2 and the maximum to 10.

Click *Deploy*.



Running one instance costs ~$45/month, so this means that this setup will cost you at least $90/month.

If the traffic volume of your website is high, Google Cloud Run will increase the number of instances, so it's possible that your setup might cost even more, e.g., 4 x $45 = $180/month.

There are also other factors that affect the final price (e.g., the volume of outgoing server requests and logging), so you might expect to pay more, e.g., $100/month instead of $90/month.

It's very difficult to estimate the monthly GCP costs of server-side tagging until you start using it.

Publish your GTM changes

Once your setup is ready, publish changes in both GTM containers (web and server). The web container will start to send your GA4 data to the server-side container, and the server container will then begin to accept the incoming data and forward it to GA servers.

Chapter III

Next steps

What to learn next?

This e-book was an introduction to the world of server-side tagging. If you want to adapt it to various use cases, you must continue learning. Not sure where to start? Here are a few ideas:

- Investigate how to optimize the costs of server-side tagging
- Learn how to send data from the server to multiple vendors (e.g., to Google and Meta at the same time). This is one of the main SGTM benefits. Ideally, you would have a small set of tracking codes client side, for example, just GA4 tracking codes. And the rest of the heavy lifting is handled on the server level (from here you can send the same event or conversion to multiple vendors). Having fewer JavaScript codes on a site will also improve the page loading speed.
- Transformations. This allows you to manipulate the incoming data and send only what's necessary to 3rd party vendors
- Changing the region of your server to EU. If your business is operating in the European Union, consider hosting the server here as well.

Things to keep in mind

Next to all the benefits and drawbacks mentioned in the previous chapters, here are more things/concerns to keep in mind.

SGTM does NOT make your knowledge of web tracking with GTM obsolete

GTM server-side does not completely replace the usual website tracking that you have been doing to this day. You still need to understand how the data layer works and how to configure tags, triggers, and variables in your web container.

Server-side tagging is an additional level of features that you *can* use in your stack. Of course, you can move entirely to the server side, but I still see many cases where the server container works hand in hand with the web container.

As the entire industry is shifting towards pure(-ish) 1st party tracking, the need for server-side will definitely increase. But you will still need to send that data from your website (and the web container in GTM is one of the options).

So, if you have doubts about reading/ watching current GTM tutorials, don't worry; feel free to dive in. If you are in a hurry and want to get up and running FAST, you can check my premium GTM courses.

Bad/shady actors

With great power comes great responsibility. And since many things are about to move to the server side, bad actors can also do that. This talk is about businesses that might/will utilize this new suite of features to avoid ad blockers, etc., and track more information without consent.

But I guess that's the price of progress. Plus, I think that the industry (and privacy guards) will continue to find ways to circumvent/limit this. Somehow.

Server-side does NOT make you automatically GDPR/CCPA/etc. compliant

Modern privacy regulations are not about technology. They are about consent. If a visitor did not give you consent for tracking, it does not matter how you collect data — client-side, server-side, whatever.

More difficult debugging

Debugging will become more complex. It will not be as simple as checking browser extensions or network requests in your browser (especially if you are debugging someone else's setup without having proper access to the GTM container).

You WILL need access to the server-side container to debug what is happening inside.

This step is the most painful for those who want to pitch new clients and showcase their expertise beforehand. Regular website tracking allows you to check what data is sent, identify issues + impress the lead. Server-side, however, is a black box that can be opened only with the proper access to the container.

Migration to SGTM

What if you have been running client-side Google Tag Manager for a while and now you want to migrate to server-side GTM? What should you do?

First, you need to make sure that your server-side setup works smoothly and your numbers will not be off too much once you make the switch.

Here's the process that I recommend. In this example, I will be talking about GA4 but similar approach can be applied to other tools/vendors too:

- Keep your live client-side setup running for a while
- Create a new (temporary) GA4 property, implement it in parallel to your live GA4 client-side setup. Basically, you will end up two sets of tags. One set sends data to your regular GA4 property, and the 2nd setup sends data to your new (temporary) GA4 property.
- Update GA4 tags of the temporary property to send data to your server-side GTM container.
- Keep this setup for a week or two. The old GA4 client-side setup continues sending data directly to GA4. Your new (temporary) setup sends data to SGTM, which sends data to a new (temporary) GA4 property. After you collect enough data in both properties, compare the numbers (such as number of sessions, events, etc.). The numbers in both properties (for the same time period) will not be identical, but they should be close (or SGTM should have better numbers).
- After you made sure that your SGTM setup works properly, remove the new setup from the website and update your old setup to start sending events to your custom server_container_url and publish all changes.

Congratulations! You have just migrated to SGTM. If you want to learn more technical details about this migration approach, I explain it in my intermediate/advanced GTM course.

How to properly learn server-side tagging

This guide was just an introduction to the world of server-side tagging. If you want to adapt it to various use cases, you must continue learning. A few topics to start:

- How to reduce the costs of server-side tagging
- Sending data from the server to multiple vendors (e.g., to Google, Google ads, Facebook CAPI at the same time)
- Manipulating data in the server
- Server-side GTM setup checklist
- Configuring your server in the EU, and much more

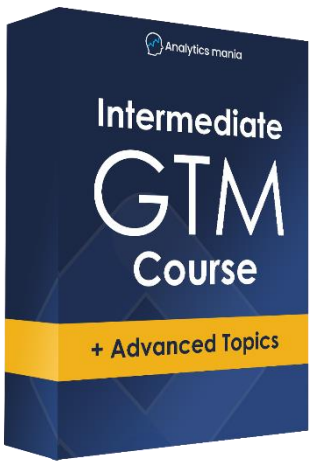
The course also contains other topics necessary for a GTM professional:

- Complete GA4 ecommerce setup
- Data Layer under the hood
- CSS selectors
- How to access data on a website when a developer is not available
- Troubleshooting and debugging your web analytics setup, and more

If you want to become a Google Tag Manager power user and benefit from it, check out my Intermediate/advanced Google Tag Manager course.

This course includes:

- 9 Modules
- Time-saving checklists, templates, etc.
- Practical tasks and a sandbox website to practice
- Lifetime 24/7 access to the course material
- Free updates
- Complete hand-holding and support
- And so much more!



Learn more about this course

Final Words

Should you start implementing server-side tagging right now in your projects?

The answer is classic “it depends”. If you want to be in control of what data is collected by marketing/analytics vendors, invest in server-side tagging.

If you want to reduce the impact on page loading speed (while doing the heaving lifting in your server), choose server-side tagging.

If any other benefit (mentioned in this article) resonated with you. SGTM is for you.

But if your analytics/marketing needs are very basic and you mainly check your data to see what are the most popular pages or where are you getting most of your traffic from, server-side GTM might be too much of an investment because it will induce additional costs.

Google Analytics and GTM have been free for years, thus many businesses got used to the fact that their tracking should be free. Spending at least 3-4 digits per year on server-side tagging might be too expensive for many of them.

This e-book was delivered to you by

Julius Fedorovicius
Founder of Analytics Mania
julius@analyticsmania.com
@fedorovicius

Analytics mania

www.analyticsmania.com/