Google Tag Manager

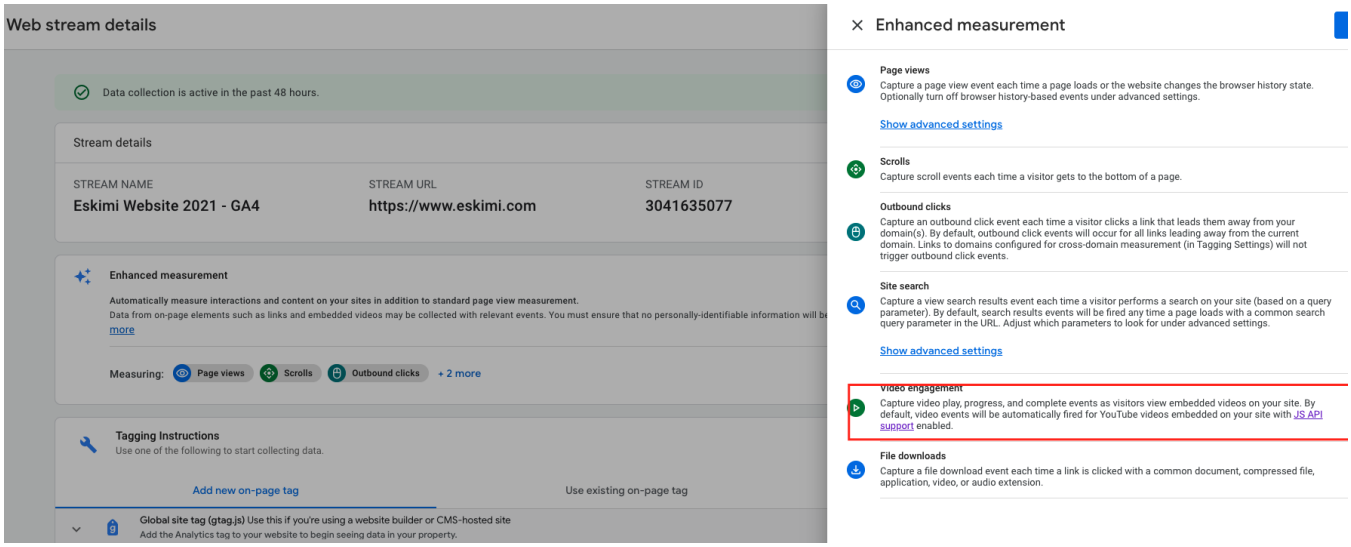
Setting up Google Tag Manager tags and tracking.

- GTM + GA4: How to Track Videos on a Website
- GTM Glossary
- GTM Definition

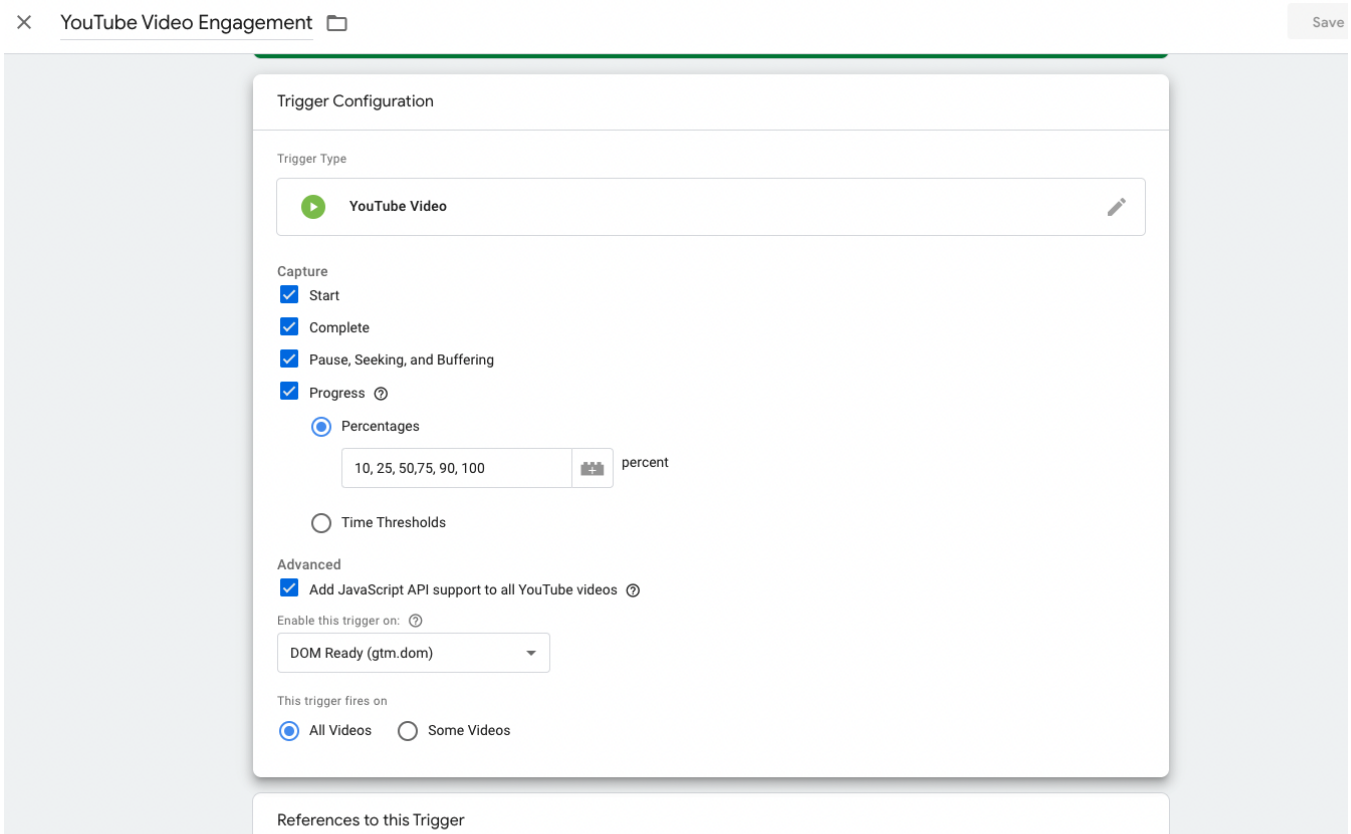
GTM + GA4: How to Track Videos on a Website

Youtube video tracking with GTM trigger and GA4

1. **Disable Video Engagement tracking features in Enhanced Measurement.**
   Otherwise, there's a high risk of occasional duplicate data.
   How to disable Video Engagement tracking in GA4:
   GA4 Admin > Data Streams > Gear Icon > disable *Video Engagement* tracking.

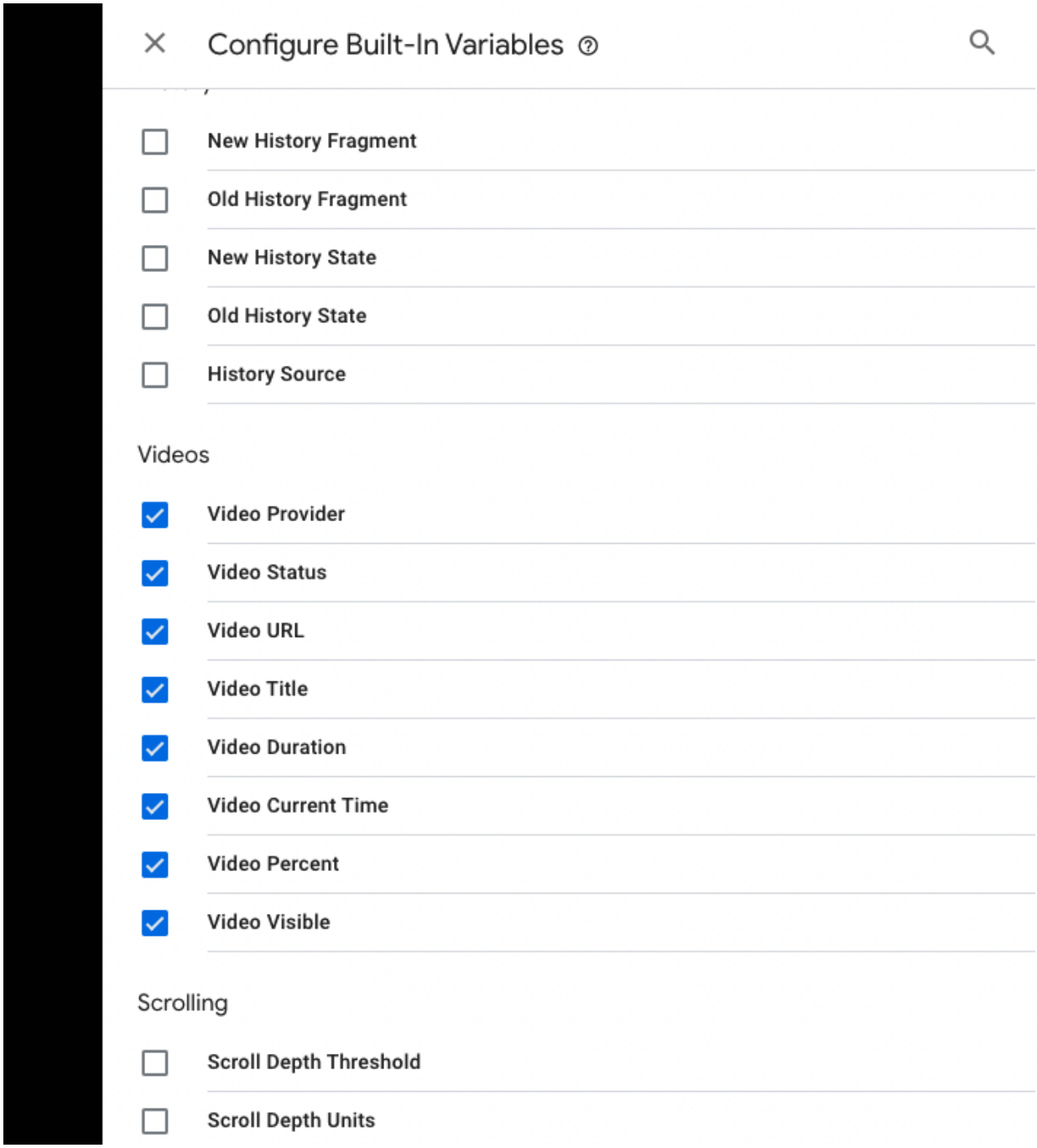


2. **GTM > Create a Youtube Video trigger**

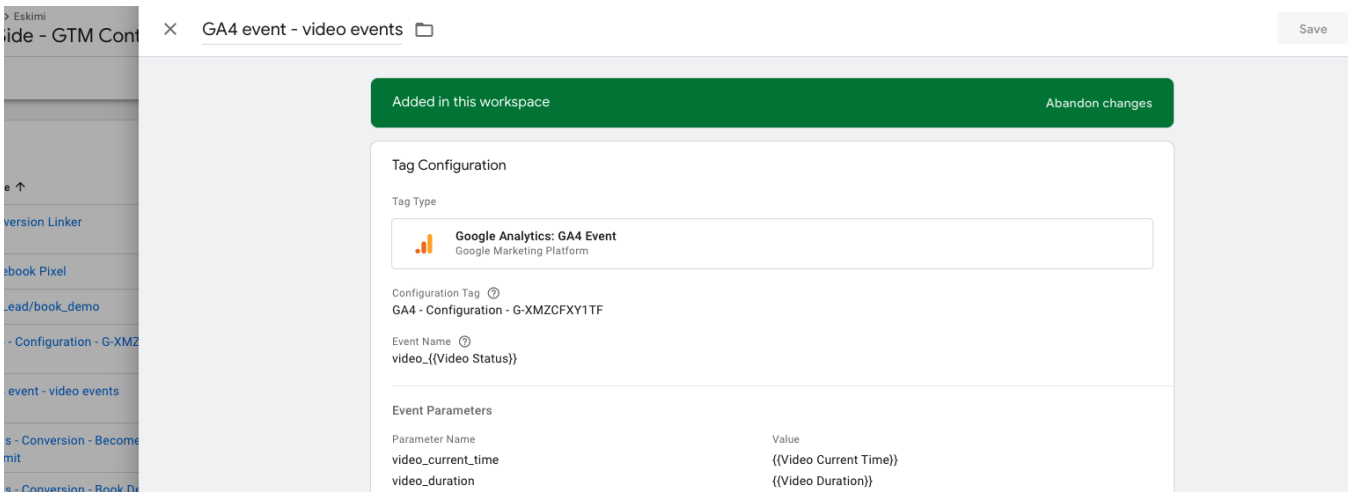


Put the details as you see it above and then hit > SAVE

3. **GTM > Enable built-in video variables**
   GTM > Variables > Configure > Select all variables that refer to videos

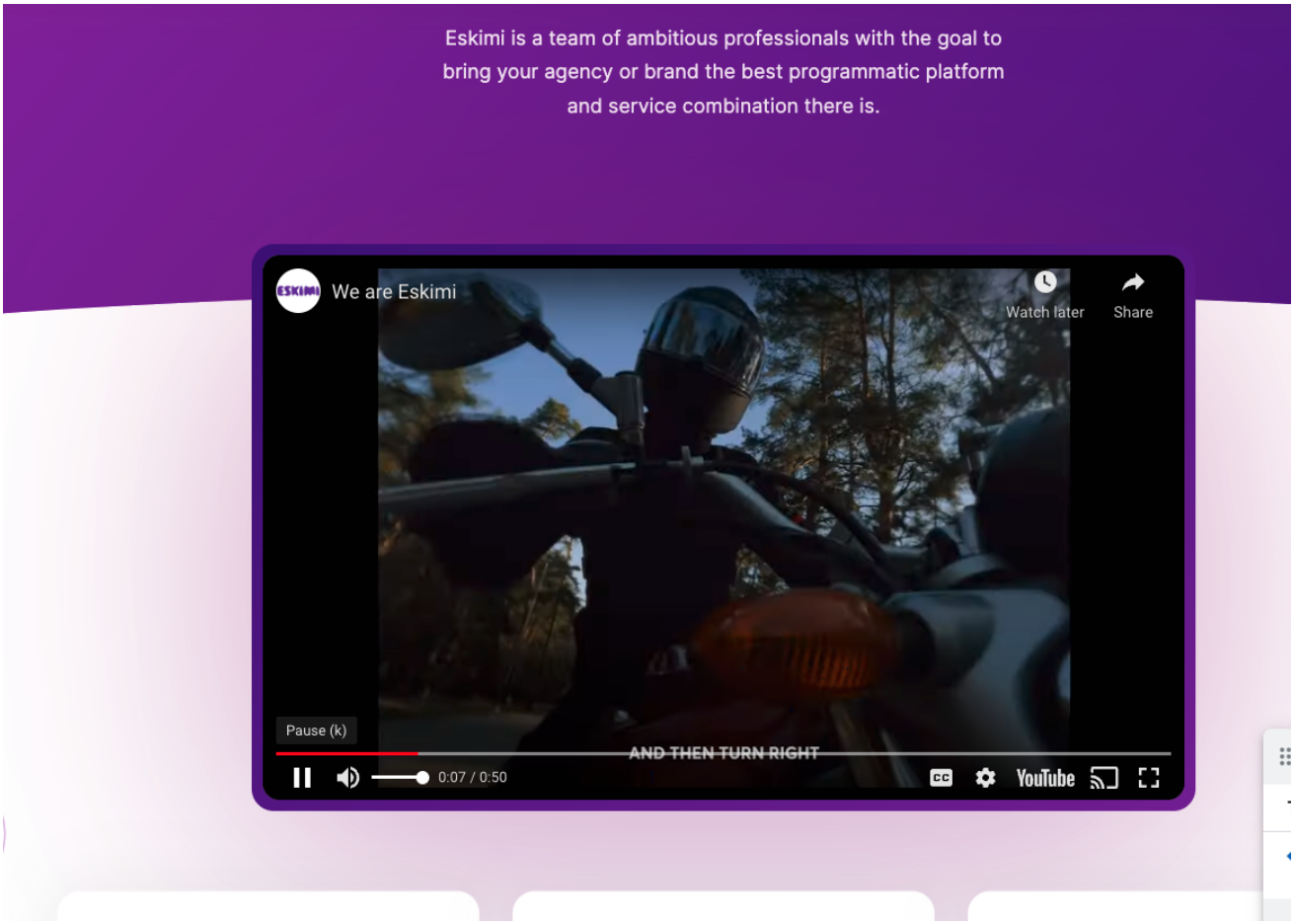


4. **GTM > Create a GA4 tag**
   GTM >Tags > New > GA4 Event Tag as you see it below

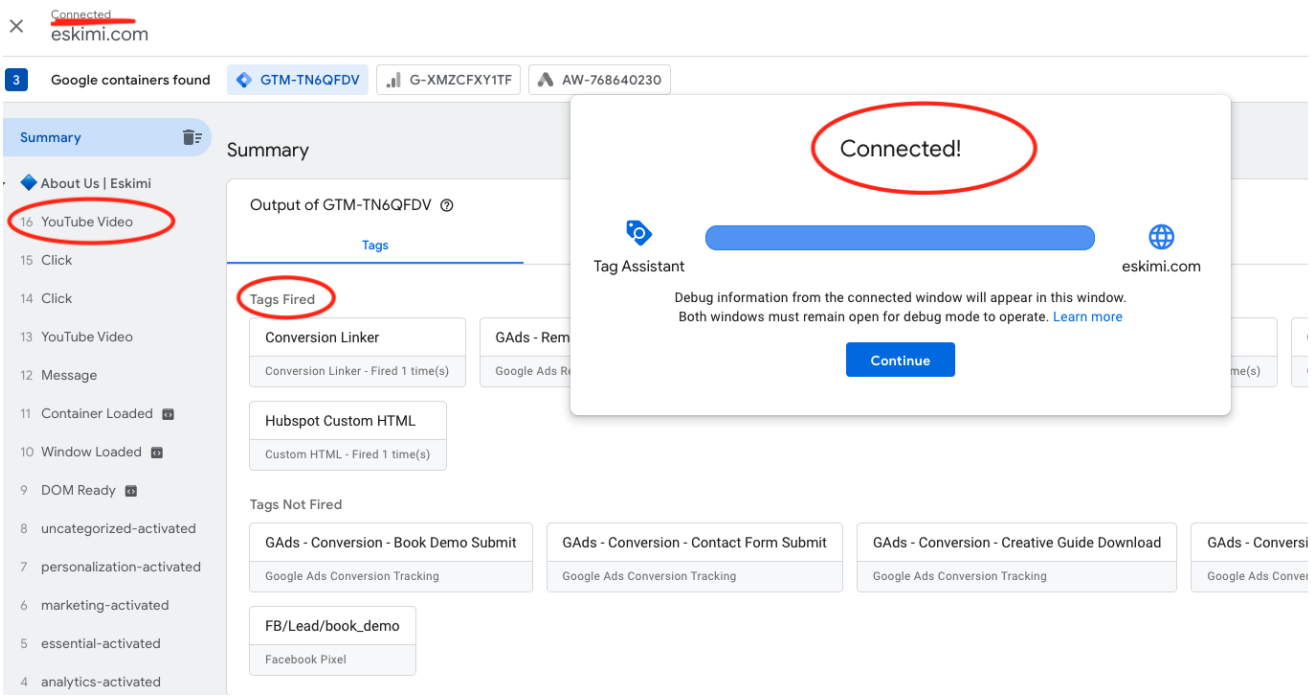


Don't forget to SAVE

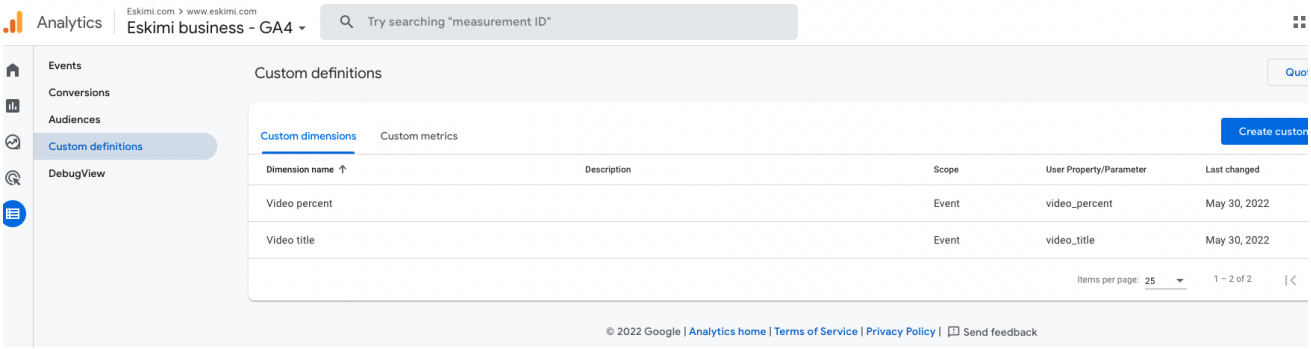
5. **Test the setup in debug view**
    1. Go to your website and find a page that contains a Youtube player, for example https://www.eskimi.com/about-us
    Then go back to GTM, refresh, select Preview and paste the link of the website.
    Choose CONNECT.
    Then, you will see that a new tab will open. Activate the video (in order to trigger the events that we just created).



Go back to GTM and see that the tag is fired :D



2. GA4 > Debug view
   GA4 > Configure > DebugView > see the events
   Also, don't forget to add the events in the Custom Dimensions [in order to be able to add these dimensions into your reports (Explorations)]



See how to make a [Custom Report for Video Engagement Metrics here.]

GTM Glossary

The Google Tag Manager terms used on these pages, in alphabetical order.

1–9

- **1st party cookie**
  A variable which returns the value for the first party browser cookie with the name you specify in the Cookie Name field.

A

- **Account**
  Google account lets you access all Google products: Gmail, Drive, Google Ads, GTM, etc.
  GA4 accounts contain *Properties.*
  Google Tag Manager accounts contain*Containers.*
- **Advanced tag settings**
  - Tag Firing Priority
  - Tag Firing Schedule
  - Tag Firing Options
  - Tag Sequencing
  - Ability to fire a tag only in published containers
- **Advertiser Tracking Enabled**
  A built-in variable in GTM containers for Mobile Apps.
  For Android, the variable returns true if ad tracking is enabled, or false if the user has opted out of interest-based ads. Learn more in the Android Developer Center. SDK versions prior to v4 will always return false.
  For iOS versions 6 and greater, the value is set to the advertisingTrackingEnabled property. Otherwise, the value is set to true.
- **AJAX**
  It stands for Asynchronous Javascript and XML.
  AJAX is a set of techniques for creating highly interactive websites and web applications.
  It enables your browser to send and receive data without refreshing the window.
  A lot of web forms are created using AJAX and standard GTM Form submission trigger cannot track it.
  That’s where AJAX listener (created by Lunametrics) comes in handy.
- **AMP**
  It’s an abbreviation of Accelerated Mobile Page.
  Simply put, Accelerated Mobile Pages is a stripped-down version of the mobile web which

runs on a reinvented version of the language used to create web pages: HTML.
This reimagined version of HTML, known as AMP HTML, removes most of the elements which cause web pages to load slower on mobile, like JavaScript and third-party scripts.
When it comes to AMP and Google Tag Manager, you need to choose a different container type – AMP.

- **API**
  API stands for Application Programming Interface, but basically, describes one way to plug your website/web application into another.
  GTM API enables developers to control accounts, containers, folders, tags, triggers, etc. via other interfaces.
- **Approval queue**
  Approval queue is a feature reserved for the DoubleClick Floodlight setup only and is used exclusively as part of the DCF implementation process.
- **Asynchronous loading**
  Unlike synchronous, asynchronous scripts can load simultaneously.
  They do not block page render and allows the browser to continue load elements while the Javascript is being downloaded in the background.
  This significantly speeds things up, especially when the JS file is loaded from a remote server.
- **Auto-event listener**
  A listener is a function which operates in the background.
  When creating the listener, you tell what operations it should wait for, and once these operations take place, the listener activates and fires any code within.
  In Google Tag Manager Web containers, default auto-event listeners listen to the following interactions:
  - Clicks.
  - Link clicks.
  - Form submissions.
  - Javascript errors.
  - Timer.
  - History changes.

  In order to enable these auto-event listeners in Google Tag Manager, you need to toggle the corresponding triggers (e.g. All Clicks trigger, etc.).
- **Auto-event variable**
  Auto-Event Variables are used to access the target element of an auto-event action (e.g. Click, Error, Form Submission).

B

- **Browser limits**
  For web containers, Tag Manager can only fire tags within the capabilities of the browser.
  Most browsers handle a maximum of 6 to 8 simultaneous HTTP requests for a single domain.
  If you have a high number of tags on the same domain firing under the same conditions,

tags will only fire within this browser limitation.

C

- **Cleanup Tag**
  In a tag, under Advanced Settings, you'll find a tag sequencing section which contains two options:
  Fire a tag before XXXX fires option, which is called a setup tag.
  Fire a tag after XXXX fires option, which is called a cleanup tag.
  A cleanup tag fires immediately after the primary tag has fired.
  To specify a cleanup tag, check the "Fire a cleanup tag when tag XXXXX is finished" checkbox and select the tag you wish to use from the menu.
  Check the "Don't fire Tag YYYYYY if Tag XXXXXX fails" checkbox if you wish to have the cleanup tag only fire when the main payload tag fires successfully.
- **Click classes**
  A built-in variable that fetches element's CSS classes upon click. The same result can be achieved by using a user-defined variable with the following settings:
  Type – Auto-event variable.
  Variable type – Element classes.
- **Click Element**
  A built-in variable that returns an HTML element that was the target of a click. You can achieve the same result by using a user-defined variable with the following settings:
  Type – Auto-event variable.
  Variable type – Element.
  Learn more about it here.
- **Click ID**
  A built-in variable that returns id value of the clicked element. You can achieve the same result by using a user-defined variable with the following settings:
  Type – Auto-event variable.
  Variable type – Element ID.
- **Click Target**
  A built-in variable which returns a string contained in the target attribute value of the clicked element. The same result can be achieved by using a user-defined variable with the following settings:
  Type – Auto-event variable.
  Variable type – Element Target.
- **Click Text**
  A built-in variable which returns the visible text inside the clicked element. You can achieve the same result by using a user-defined variable with the following settings:
  Type – Auto-event variable.
  Variable type – Element Text.
- **Click – All clicks**
  A trigger which fires when a visitor clicks anything on the page (image, link, etc.). In Preview and Debug console, this event is displayed as gtm.click.

- **Click – Just clicks**
  A trigger which fires when a visitor clicks any link on the page. It listens only for clicks which propagate up to a link (<a/>) node. This means that you can click on a SPAN in a BUTTON in a DIV, but as long as there's a link somewhere up there, GTM will register the event as a link click.
  In Preview and Debug console, this event is displayed as gtm.linkClick.
- **Click URL**
  A built-in variable which returns the value of href (or action) attribute of the clicked element. You can achieve the same result by using a user-defined variable with the following settings:
  Type – Auto-event variable
  Variable type – Element URL
- **Constant**
  A user-defined variable which returns the string you choose to type in the Value field. Its value never changes. Learn more about its use cases.
- **Container**
  Every Google Tag Manager account contains one or more containers. Currently, there are 5 types available: Web, iOS, Android, AMP, or Server. Each container has one container snippet.
- **Container Activity**
  Container activity displays information about the actions that have been done in the container. It is useful to determine which user has performed a certain action and when it was done.
  You can find container activity by opening any container of the account you're interested in > Admin > Container Activity.
- **Container ID**
  Container ID is a part of Google Tag Manager Container snippet.
  e.g. GTM-1000AB
- **Container Snippet**
  The Google Tag Manager container snippet is a small piece of JavaScript and non-JavaScript code that we need to paste into website's pages.
  It enables Tag Manager to fire tags by inserting gtm.js into the page (or through the use of an iframe when JavaScript isn't available).
  Without this code, Google Tag Manager would not work on the website.
- **Container Snippet Placement**
  Container snippet consists of two parts: <script> and <noscript>.
  It's highly recommended to copy and paste the <script> as close to the opening <head> part as possible on every page of the website.
  <noscript> should be placed immediately after the opening <body> tag of the page.
  While we can place both these codes in <body> only, we must not add them both in <head>.
  The <noscript> part renders an iframe, which is not allowed in <head>, otherwise the website's HTML will be invalid.
- **Container Version**
  A built-in variable which returns the version number of the container that is implemented

on the site, or QUICK_PREVIEW if the version is in preview mode.
The same result occurs by utilizing a user-defined variable - Container Version Number (no configuration is required).

- **Cross-Domain Tracking**
  Google Analytics can use only the cookie created by the domain itself, and by default, each domain is separate from the other domains.
  Each domain exists on their own, and therefore, cannot share information (by default).
  Cross-domain tracking makes it possible for Analytics to see sessions on those two related sites as a single session.
- **Custom Auto-Event Listener**
  A custom auto-event listener is a function which operates in the background which is waiting for the particular actions occur on the website.
  When the listener activates it pushes certain data to the Data Layer.
  In addition to the default auto-event listeners, we can easily add custom listeners to track interactions, for example, Facebook LIKE button clicks, Vimeo player, AJAX form submissions, etc.

G

- **Google Ads Conversion Tracking**
  A tag type which lets you track Google Ads conversions. Usually, this tag fires when a visitor/user completes registration, purchase or any other important action.
  To start a new Google Ads tag, from the home screen select New Tag → Tag Configuration → Google Ads Conversion Tracking.
- **Google Ads Remarketing Tag**
  A tag type which lets you install Google Ads remarketing code.
  This enables you to show ads to people who have visited your desktop or mobile website, or (did not) complete a particular action.
  To create a new Google Ads tag, from the home screen select New Tag → Tag Configuration → Google Ads Remarketing.

GTM Definition

What is GTM (Google Tag Manager)?

GTM is a tag management solution which acts as an intermediary between a website and 3rd party tracking tools.
You need to add the tracking codes to GTM and then configure the rules when they should fire.

Why using it?

1) With GTM we can test the tags to make sure they are triggered when we load a page or click a particular button.
2) We can change the tags without changing the website's source code.
3) GTM lets us manage various JavaScript tracking codes on a website.
4) Fast Deployment of Tracking Codes
5) All Tags in one place
6) Testing and debug tools
7) Tag templates > you don't have to setup everything from scratch
8) HUGE active community on FB
9) Versions of the container (you can restore everything)
10) Dynamic Tagging + Privacy Policies

How it works?

A **tag** is a piece of code that fires on a website under certain circumstances.
It can be a tracking code, some piece of code that changes the text or a particular website element, or even a code which changes the color of the browser's address bar, etc.
When you create a tag, you basically instruct Google Tag Manager to "do this", "do that", "track page views of this visitor", "track this click and send to Google Analytics" etc.
A **trigger** is a condition when a tag must fire.
Should a tag fire on all page views? Or maybe on certain clicks?
How about successful form submissions?
All these examples are triggers.
When a particular condition (or a set of conditions) is met, a trigger is activated and all the tags (linked to it) are dispatched.
A **variable** is the final member of this trinity.
Variables are little helpers that can be used in tags, triggers, or even in other variables.
A variable can:
■ Hold a single piece of data (like page URL, website domain, product ID, text of a link, etc.)

■ Hold a set of data/settings (Universal Analytics Google Analytics settings variable contains multiple settings related to GA, like Tracking ID, Display Advertising settings, etc.)

