rudderstack

The Definitive Guide

Overcoming the limitations of GA4

Overcoming the Limitations of GA4

Table of Contents

In today's data-driven world, organizations need a clear understanding of their customers to gain competitive advantage. Web analytics are an important piece of building this understanding as they deliver valuable insights into how users interact with a company's digital assets.

Google Analytics 4 (GA4) is the latest version of Google's web analytics platform. It introduces several new features and changes compared to its predecessor, Universal Analytics (UA). GA4 introduces an event-based data model, enhanced tracking capabilities, and deeper integration with Google's ecosystem.

The move to a modern, event-based data model is the most significant change and requires reeducation and planning, but another change could significantly affect your business that's important not to overlook: How GA4 supports server-side tracking.

SERVER-SIDE TRACKING BENEFITS

As the analytics landscape evolves, it's getting more difficult to collect complete data for conversions and attribution. This is largely driven by two factors:

- Page speed - Each second delay in desktop or mobile page load was approximated to reduce conversion rates between 20-22% percent
- Ad blockers - Over 40% of people worldwide now use ad blockers

If you're relying solely on client-side tracking, it's likely your conversion rate is suffering and you're losing a significant amount of valuable data to ad blockers. Server-side tracking can help you stay competitive in the face of these challenges because it delivers:

- Faster website performance – server-side tracking reduces the amount of code required to load client-side, which can lead to dramatic improvements in page speed metrics.
- Ad blocker resiliency – Server-side tracking enables you to capture data despite ad blockers, so you can regain large volumes of data that were previously unavailable.

Perhaps most importantly, server-side tracking lets you own your first-party data capture. Rather than injecting third-party code throughout the site, you can send data server-side to command more granular control over what happens with the data.

Challenges with GA4

Compared to Universal Analytics, GA4's server-side tracking is less independent and requires support from client-side tracking to enable critical functions that you may depend on. The functionality that requires client-side tracking includes attribution, sessionization, geolocation, and conversions for Google Ads remarketing.

This forces a painful tradeoff. If you rely solely on server-side tracking, you won't have access to key functionality. If you rely solely on client-side tracking, your site performance will suffer, and you'll have to deal with excessive data loss because client-side tracking is notoriously susceptible to ad blockers.

This guide explores two steps to address these challenges. By **adopting a hybrid approach to GA4** using RudderStack to easily enable both server-side and client-side tracking, you can get the best of both worlds and unlock GA4s full power. Eventually, you may also want to go beyond GA4 to answer harder questions. For teams interested in this second step, we'll cover how to **implement a warehouse analytics strategy** to help your company gain more data ownership, future-proof your analytics, and improve integration with the rest of your technology stack..

GA4 SERVER-SIDE TRACKING OPTIONS

GA4 offers two options for server-side tracking: Google Tag Manager (GTM) Server-side or the GA4 Measurement Protocol API. We'll detail both options and their limitations below.

Google Tag Manager server-side

GA4's GTM server-side tagging is different from a web GTM container. It's hosted on a server-side container (GCP or other), so it does move instrumentation off of the browser, but it comes with some critical limitations:

It's not 'pure server-side' without browser code because it still requires a web GTM container to send events to the server-side container

Browser GA4 tags are used like a server-side data layer, and data is transformed to other tags (Facebook, Snapchat, etc.). This data mapping can introduce extra points of failure for some sites.

For example, if your web development team maintains GTM's data layer for event tracking, GTM server-side cannot natively retrieve this data. Instead, GTM server-side retrieves data that passes through the GA4 tagging, which can introduce discrepancies compared to pulling from the raw data layer.

GA4 Measurement Protocol API

The measurement protocol API enables you to write code and send data directly to https://www.google-analytics.com/mp/collect. It's an effective way to augment client-side data collection, and it's the method RudderStack uses to send GA4 server-side data, but its standard implementation is problematic:

- GA4's out-of-the-box attribution reporting does not support server-side data, it requires browser tracking.
- Google Ads remarketing requires browser tracking.
- Out-of-the-box geographic information requires browser tracking.
- Sessionization requires a session ID, client ID, and timestamp that fits within a particular session, and this implementation requires careful QA.
- Engineering a server-side client ID for anonymous users can require new development work for some businesses.

Currently, using only the Measurement Protocol API for data collection eliminates critical capabilities of GA4 reporting. GTM server-side tagging works around this because it still requires a web container or Google tag (gtag.js) coded in the site, but as noted, this approach comes with its own limitations.

Because both server-side approaches come with significant drawbacks, most organizations will require a hybrid GA4 implementation that includes server-side tracking with support from browser code.

ARCHITECTING A HYBRID DEPLOYMENT

Hybrid deployment delivers the best of both worlds while capturing a fuller set of information. Key benefits of hybrid deployment:

- Attribution can be created and updated via browser code (GTM, gtag.js)
- Sessionization can be generated through browser code (GTM, gtag.js)
- Geolocation data can be captured through browser code (GTM, gtag.js)

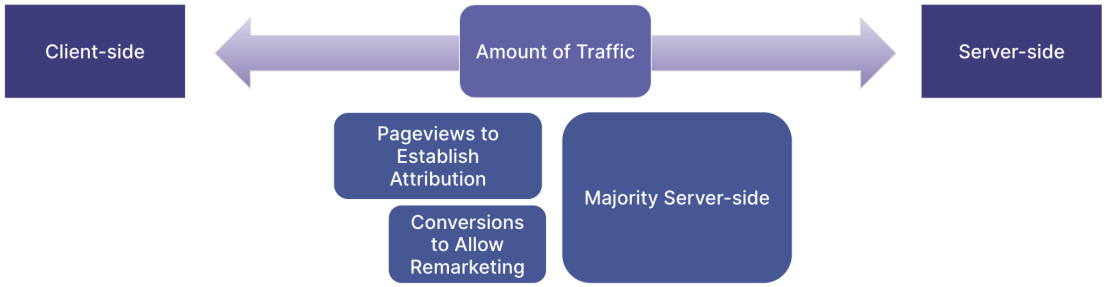
- Conversions for Google Ads Remarketing can be captured through browser code (GTM, gtag.js)

For many companies, these benefits are non-negotiable because they enable capabilities their teams have come to rely on. There are some drawbacks to hybrid deployments, however, that should be considered. Hybrid deployments are more time consuming to set up, they do still require client-side code that affects performance, and they are partially susceptible to ad blockers.

There are three primary ways to set up a hybrid deployment, and each option involves its own tradeoffs. We'll illustrate this with a few examples below.

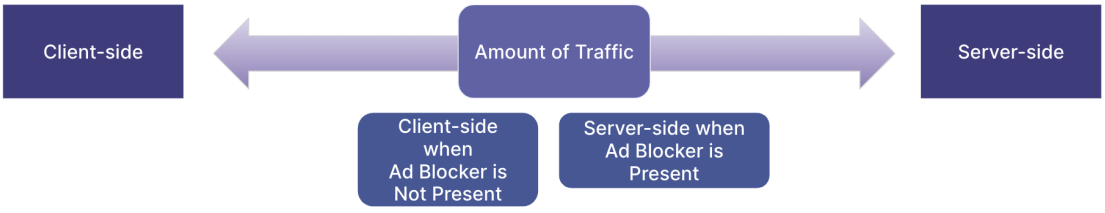
Option 1: Server-side with client-side as needed

This deployment uses server-side tracking everywhere that client-side tracking isn't necessary for attribution, remarketing, or geolocation data. It has the least impact on client-side performance and reduces data loss from ad blockers, but it requires a significant organizational commitment to server-side tracking that necessitates training and upkeep.



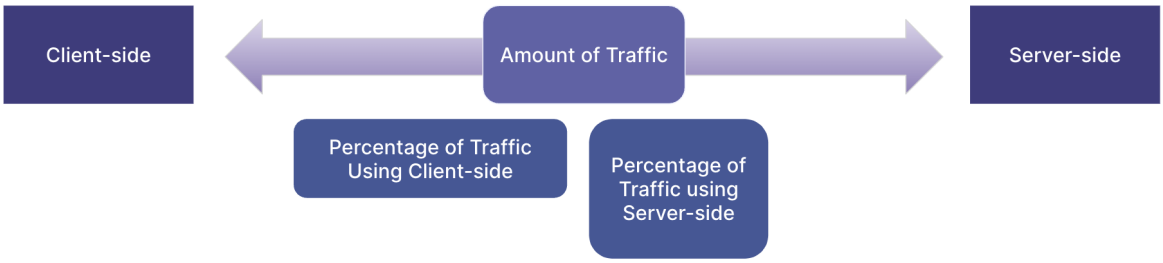
Option 2: Server-side fallback for ad blockers

In this setup, server-side tracking is used only when a client-side ad blocker is detected. It reduces data loss from ad blockers but still involves client-side impacts on performance.



Option 3: Split server-side and client-side

Splitting server-side and client-side tracking runs both tracking methods in parallel. Data is then deduplicated in reporting. When a matching client-side hit cannot be found for a particular server-side hit, then the positive impact of server-side performance is demonstrated.



This approach allows organizations to directly compare client-side data collection with server-side data collection, and because client-side tracking is still fully running, it provides you with more time to train your team and update server-side tracking. As you can gather, though, it requires reporting resources for de-duplication and still involves client-side impacts on performance.

With GA4, a hybrid implementation is the only way to access a fuller, more unified, and accurate set of attribution data with the least impact on performance. But setting up a hybrid implementation with the tools Google provides is complicated and requires maintenance. That's why we added a hybrid deployment option to RudderStack's GA4 integration.

USING RUDDERSTACK HYBRID MODE

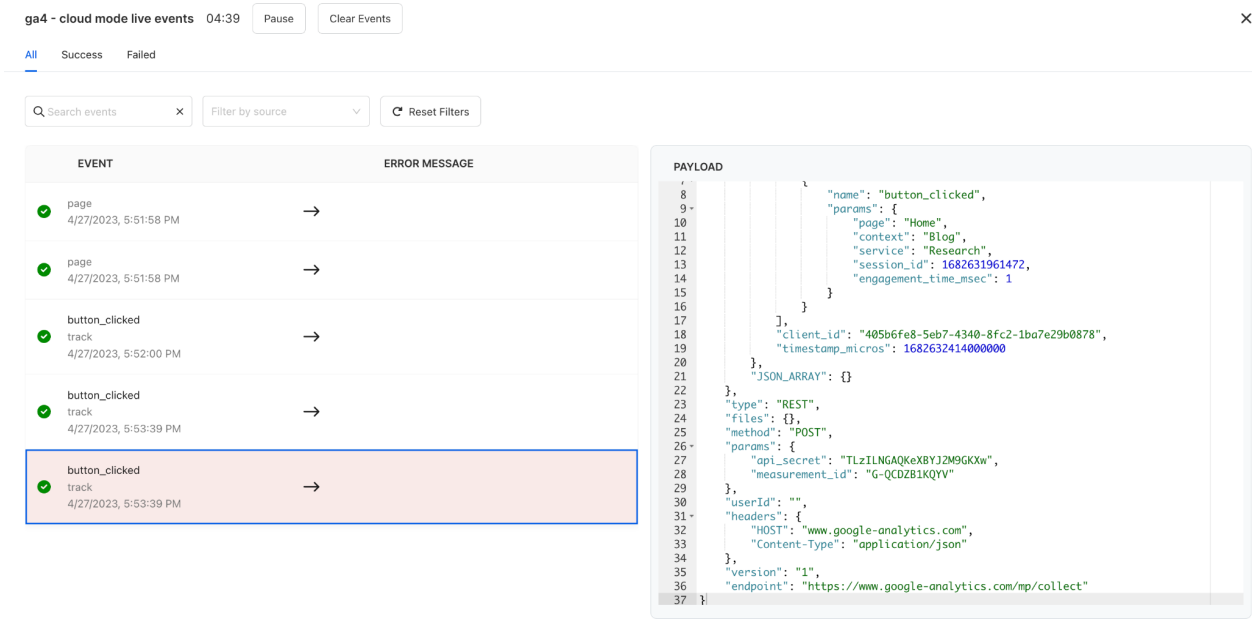
RudderStack is a warehouse-native customer data platform that facilitates the direct transfer of user behavioral events to and from your data warehouse. Additionally, it provides an integration library that supports the transmission of data to over 180 other tools. These tools include popular analytics platforms like Google Analytics 4.

RudderStack's GA4 destination supports both a cloud and a device mode connection. Here, we'll highlight the main differences between them.

Device mode loads the GA4 SDK (gtag) directly on a website. It's designed for **client-side** connections where events do not pass through RudderStack's server. This type of connection supports the complete range of features available in GA4, but it's vulnerable to ad blockers and can severely impact site performance.

data:image/svg+xml;…	200	svg+xml
rudder-analytics.min.js	200	script
?p=cdn&v=2.32.0&writeKey=2HxrdF4Y9oKmM2pmjQmTB9i…	200	xhr
page	200	xhr
GA4.min.js	200	script
js?id=G-TVZYF9K3L4	200	script
js?id=G-Q56X1VG5L6&l=dataLayer&cx=c	200	script

Cloud mode makes requests to Google's Measurement Protocol API, which represents the GA4 **server-side** connection. This allows developers to directly send pageviews and other user interaction events to the GA4 servers via HTTP requests, thereby enabling the tracking of user behavior across various platforms, including websites and mobile apps, as well as offline events that aren't generated within a web app. In contrast to device mode, cloud mode events are processed through the RudderStack server and can be viewed in RudderStack's Live Events viewer.



The cloud mode connection delivers the benefits of server-side tracking covered above, but as noted, key GA4 functionality does not support server-side data, so relying solely on cloud mode for analytics reporting won't provide a full picture and hampers GA4s functionality. Google even state's that server-side tracking is meant to augment the client-side gtag configuration, not to fully replace it.

To make it easier for teams to reap the advantages of both connection modes, RudderStack's GA4 integration now includes a third connection mode: hybrid mode. This new feature produces a one-step GA4 hybrid deployment that automatically blends information from the client-side and server-side connections.

What happens in hybrid mode:

Choosing the Hybrid mode option divides events into two streams:

- **Device mode** captures Page events.
- **Cloud mode** captures Track and Identify events.

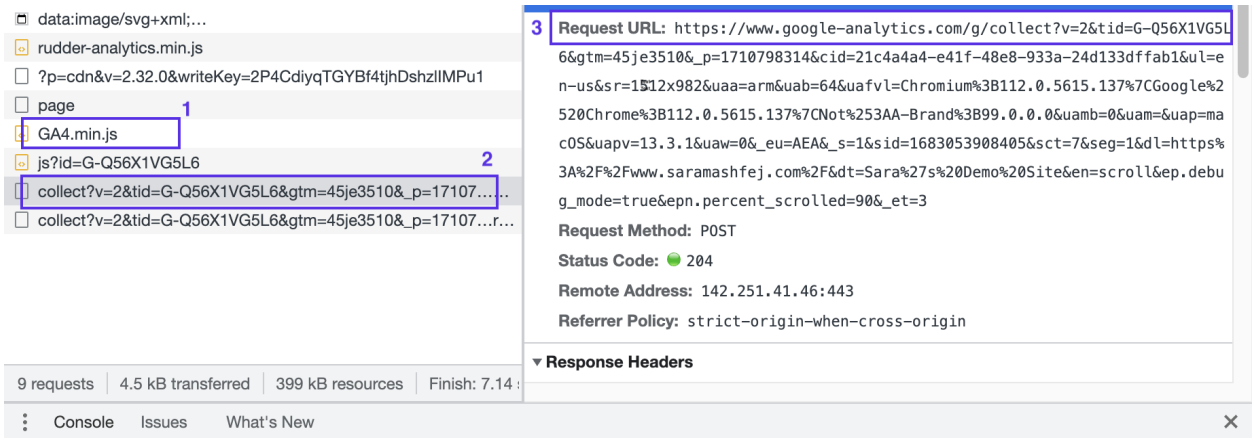
On the RudderStack backend, we grab the session ID and client ID from gtag events sent through device mode and merge them with events captured through cloud mode to build accurate and full reporting of user interactions within that session.

This configuration improves site speed because it reduces network calls by only capturing Page calls with device mode (client side) which supports remarketing, UTM, and geolocation tracking, while Track calls containing crucial data are captured faster and with less data loss by leveraging Cloud mode (server-side).

Hybrid mode benefits

RudderStack's GA4 Hybrid Mode connection gives you access to an implementation that actually does deliver the best of both worlds. You'll get a fuller, more unified, and accurate set of attribution data with the least possible impact on front-end performance through a combination of client-side and server-side tracking. This feature follows the approach detailed above in option one – the server-side with client-side as needed option.

Here's an example of a GA4 Hybrid configuration running on a website:



1. Refers to the gtag being loaded to the site: this represents the client-side connection.
2. Request being made to the Measurement Protocol API: this represents the server-side call being made

3. The Request URL that represents the Measurement Protocol endpoint where the events will be sent to: https://www.google-analytics.com/g/collect along with the query parameters.

This feature also makes it easy to get a GA4 hybrid deployment up and running. With RudderStack's integration, you don't have to create two separate connections and then de-duplicate events via filtering and transformations. You can simply set up one connection and flip on hybrid mode.

Check out the docs and this knowledge base article on How to Send Data to GA4 Hybrid Mode to learn how to get started with RudderStack's GA4 Hybrid Mode.

With Rudderstack, you can get full value out of GA4, however for more sophisticated analytics, you might still find GA4 limiting. Its core model doesn't allow for much flexibility or data ownership, and there are simply questions you can't answer with GA4 alone.

Fortunately, with RudderStack, when you run up against GA4's fundamental limitations, you already have the data layer setup to enable warehouse analytics. Using warehouse analytics, you can build robust analytics that provide better insights into the entire user journey and are consistent across the entire data stack.

GOING BEYOND GA4

Many data and marketing teams are going beyond Google Analytics. GA4 is a better tool than UA in many ways, but due to its core model, limitations persist even with a hybrid implementation.

The benefits of owning your analytics infrastructure

Thankfully, you don't have to subject your data team, or any other team, to the limitations of Google Analytics. Using RudderStack for first-party data collection and your warehouse for storing, modeling, and serving data, you can build a future-proof analytics stack that will deliver data you can trust and easily scale with your business.

> "*Data has changed, and for those still depending on Google Analytics, this is the perfect moment to modernize your data stack. Your customer data is the most valuable asset your business has, and you need to own all of the data and stop being dependent on tools like GA to host or report it. With a solid CDP, you can now track all your customer data and pipe it to a warehouse, and own your data. The businesses that own their data in the future will be the ones who win*"
>
> *Dan McGaw*
> *Founder & CEO of McGaw.io, tech stack and analytics experts*

Analytics with infinite optionality

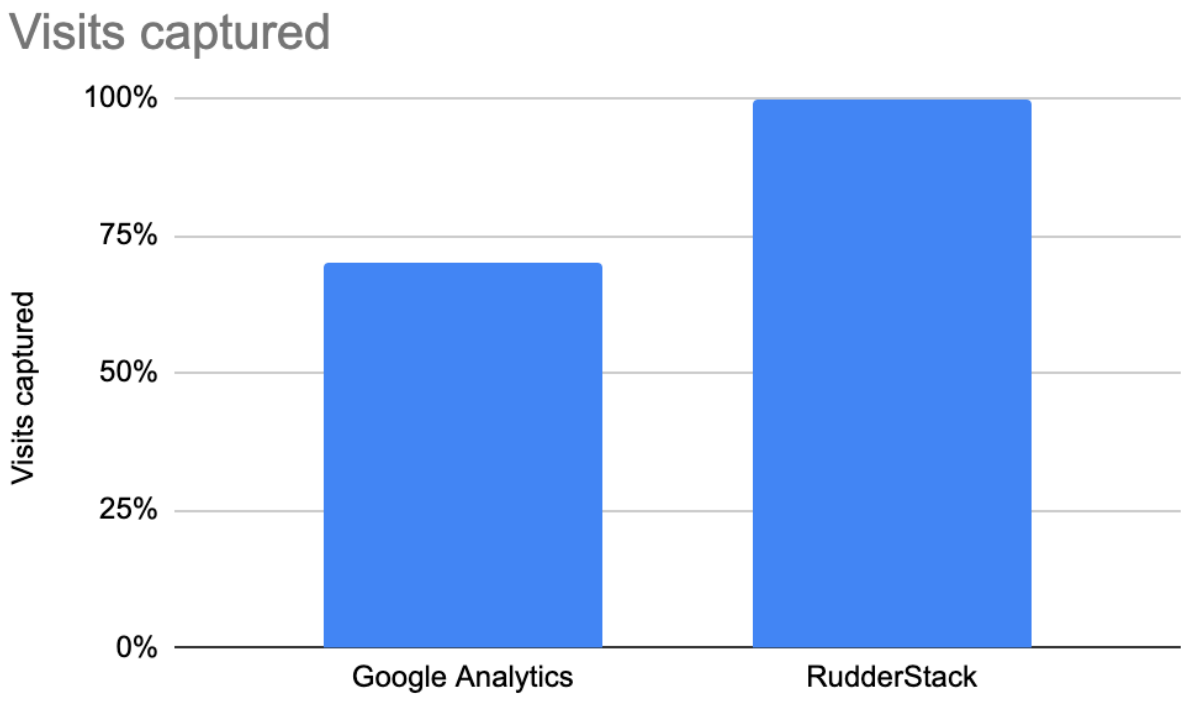
Abstracting your data capture infrastructure away from your data storage and transformation layers, as well as the analytics tools themselves, gives you the ability to modify any individual component of your stack as the analytics needs of your business change.

Do your A/B testing or product teams want to try a new analytics tool? No problem—just point the event stream at a new destination without opening any dev tickets (more on this later). Do your analysts need to modify metrics to reflect a new pricing plan? You can move fast by updating the queries running on your warehouse without having to completely overhaul event instrumentation and your entire visualization layer. What if your data science team is building a new recommendations model on a fresh data lake partition? You can avoid painful and slow batch jobs by simply adding a new data store as a destination for your existing events.

Owning your analytics infrastructure means giving the data team full control and flexibility to quickly adapt to the needs of the business without facing the painful limitations of vendor lock-in.

Capture every site visit

Capturing all of your data requires using an analytics-agnostic tool specifically designed to collect raw, first-party data. RudderStack's SDKs are purpose-built for capturing customer events and can be proxied behind your site or app URL, giving you full data capture.



See the whole picture with rich, configurable, transformable payloads

RudderStack's out-of-the-box payloads provide far richer data than GA4. They also feature a standardized, open-source JSON schema with objects you can customize with both event properties and user traits that meet the specific needs of your business. If analytics needs change down the line, which they always do, you can use RudderStack's transformations to rename keys or reshape payloads without having to touch the code in your app or website.

GA4 page payload

```
v: 2
tid: G-252450485
gtm: 2oebu0
_p: 309274267
cid: 679471488.1651608029
ul: en-us
sr: 2560x1440
uaa:
arm
uab: 64
uafvl:
Google%20Chrome;107.0.5304.121|Chromium;1
07.0.5304.121|Not%3DA%3FBrand;24.0.0.0
uamb: 0
uam:
uap: macOS
uapv: 12.6.0
uaw: 0
_s: 2
dl: /pricing/
dr: $direct
dt: RudderStack Pricing
sid: 1671219066
sct: 1
seg: 1
en: page_view
_ee: 1
```

RudderStack page payload

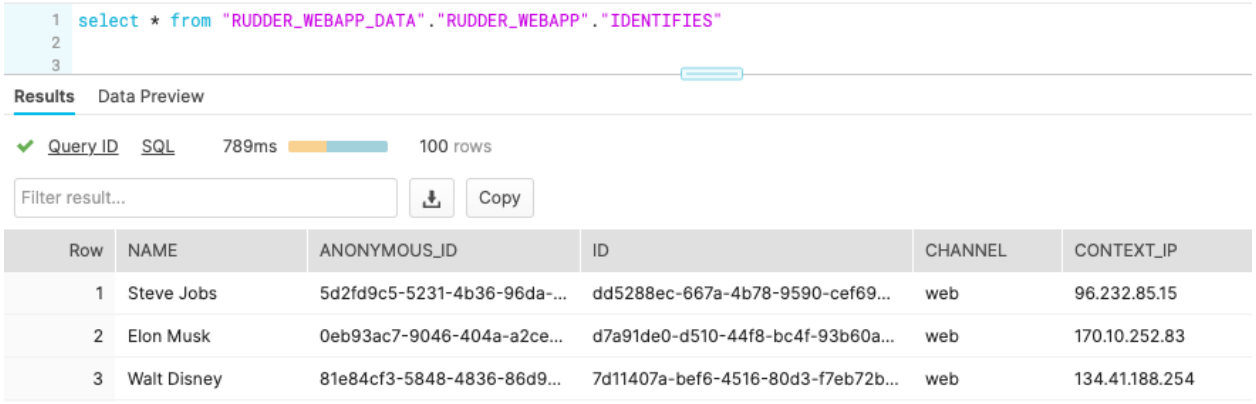
```
{
    "channel": "web",
    "context": {
        "app": {
            "build": "1.0.0",
            "name": "RudderLabs JavaScript
SDK",
            "namespace":
"com.rudderlabs.javascript",
            "version": "1.21.0"
        },
        "traits": {
            "name": "First Name"
        },
        "library": {
            "name": "RudderLabs JavaScript
SDK",
            "version": "1.21.0"
        },
        "userAgent": "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/107.0.0.0 Safari/537.36",
        "locale": "en-US",
        "os": {
            "name": "",
            "version": ""
        },
        "screen": {
            "density": 2,
            "width": 2560,
            "height": 1440,
            "innerWidth": 1280,
            "innerHeight": 787
```

```
        },
        "sessionId": 1671218225671,
        "campaign": {},
        "page": {
            "path": "/pricing/",
            "referrer": "$direct",
            "referring_domain": "",
            "search": "",
            "title": "RudderStack pricing",
            "url":
"https://rudderstack.com/pricing/",
            "tab_url":
"https://rudderstack.com/pricing/",
            "initial_referrer": "$direct",
            "initial_referring_domain": ""
        }
    },
    "type": "page",
    "messageId":
"1671218575125100-c1271cc4-5b0c-4b99-92a7-58
cb69310802",
    "originalTimestamp":
"2022-12-16T19:22:55.125Z",
    "anonymousId":
"8bd1d069-0318-4d97-bfca-6b4a39ab0874",
    "userId": "eric@rudderstack.com",
    "properties": {
        "path": "/pricing/",
        "referrer": "$direct",
        "referring_domain": "",
        "search": "",
        "title": "RudderStack Pricing",
        "url":
"https://rudderstack.com/pricing/",
        "tab_url":
"https://rudderstack.com/pricing/",
        "initial_referrer": "$direct",
        "initial_referring_domain": ""
    },
    "integrations": {
        "All": true
    },
    "sentAt": "2022-12-16T19:22:55.135Z"
}
```

Build confidence with full transparency in your warehouse

When you store and model all of your data in your warehouse, you never have to wonder what GA4 is doing inside of the black box and whether their algorithms are making the right choice for your business.

> *"We were using the free version of Google Analytics. Everything was anonymized, and we couldn't see what our users were doing on our website or how they were using our product. RudderStack has increased visibility into user behavior and user journeys, given us deeper insight into our funnel and we can run A/B tests that let us customize the user experience."*
>
> *Mona Sami, Director of Data Analytics, InfluxData*



```
1  select * from "RUDDER_WEBAPP_DATA"."RUDDER_WEBAPP"."IDENTIFIES"
2
3
```

Results | Data Preview

Query ID | SQL | 789ms | 100 rows

Filter result... | Copy

Row	NAME	ANONYMOUS_ID	ID	CHANNEL	CONTEXT_IP
1	Steve Jobs	5d2fd9c5-5231-4b36-96da-...	dd5288ec-667a-4b78-9590-cef69...	web	96.232.85.15
2	Elon Musk	0eb93ac7-9046-404a-a2ce...	d7a91de0-d510-44f8-bc4f-93b60a...	web	170.10.252.83
3	Walt Disney	81e84cf3-5848-4836-86d9...	7d11407a-bef6-4516-80d3-f7eb72b...	web	134.41.188.254

Building analytics on your warehouse gives you full transparency and flexibility

Deploy and modify flexible data models that match actual behavior

Using tools like dbt, you can build your own models for key use cases identity resolution and sessionization, then modify them as you learn more about how your users interact with your website and apps.

Break free from Google's compliance chaos

Perhaps most importantly, owning your analytics infrastructure means you can say goodbye to Google's compliance chaos once and for all, meaning your infosec, legal and marketing teams can rest easy.

HOW TO GUIDE: GETTING STARTED WITH WAREHOUSE-FIRST ANALYTICS

In this section of the guide, we'll cover the definition of warehouse-first analytics and walk through the steps of replacing GA4 with your own data infrastructure.

What are "warehouse-first" analytics?

A warehouse-first data analytics stack is an analytics stack with a single data repository (otherwise known as the data warehouse) for all customer data. As we mentioned above, analytics tools like GA4 create data silos, but so do other customer data repositories like Salesforce. This makes it challenging for organizations to access all the data they need to make complex decisions.

By gathering all of your company's data into a single location, including traffic data from your websites and apps, you're able to build efficient analytics on large, diverse, high-quality datasets to answer questions that analytics tools like GA4 are not able to answer on their own.

For example, suppose your company is a startup in its early stages. In that case, you're probably just trying to understand how customers perceive or interact with your product, so you likely have questions like:

- How many active users did we have this month?
- What's our overall retention rate?
- What are our top sources of referral signups?

For those basic questions, GA4 can get the job done (albeit with limited, siloed data), but as your company grows, you'll need answers to more complex questions like:

- What behavior in a user's first day indicates they're likely to sign up for a paid plan?
- How does lifetime value vary by paid advertising channel?
- What patterns lead to an increase in purchases of our products?
- What products should we recommend to specific users to increase revenue?
- How should we price our product based on the cost of acquisition or cost of resources per customer?

You can't answer those questions with GA4. But you can with warehouse-first analytics. If you have all of the data in your warehouse, your data and analytics teams can not only model the raw data to find the answers, but understand how those answers change over time—all with fully transparent, version-controllable queries.

The components of a warehouse-first data analytics stack

A warehouse-first data analytics stack is made up of a few crucial components to facilitate the collection, storage, and movement of your data to make it easier to analyze:

1. Cloud data warehouse
2. Behavioral data ingestion
3. Data modeling
4. Visualization

For each element, let's look at a few tools fit to get the job done.

1. Centralizing the data: your data warehouse

With raw traffic and behavioral data—along with all of your other customer data—in a single place you get complete ownership of your data, plus you can perform much deeper analyses. It also means you can easily move between and onboard different analytics tools for different teams because each tool has access to all of your historical data.

Numerous data warehouse options are available, but Redshift, Snowflake, and BigQuery have the most adoption. While each of these warehouses has its own unique advantages, they all work well for ingesting event payloads from a tool like RudderStack to drive web analytics. Warehouses are used for more than web analytics, though, so data teams will want to consider other questions like the additional types of data being stored (i.e., semi-structured data), how much performance management is required by engineers, and how the cost scales depending on your company's use case.

2. Collecting the data: event ingestion

Once you have a centralized store for your data, you need to start collecting it. Ingestion is a massive topic when it comes to getting data into your warehouse, so in this guide, we're going to focus on what you need in order to replace GA4 for web and mobile analytics.

To build web and mobile analytics, you need to capture user behavior data, which comes in the form of both *events* and *user traits*. **Events** describe a user action, like making a purchase, while **traits** describe the individual user who performed that action.

As modern data teams know all too well, Google Analytics is a notorious silo for this kind of behavioral data and, even though you can technically export the data through ETL jobs or, with GA4 to Google products, it still requires a ton of modeling to be usable...and that's before you even try to figure out how to join user identities with other kinds of customer data from other sources.

All of these problems are made worse when additional scripts from other analytics tools are added to the equation, creating multiple different dedicated ingestion pipelines that feed their own silos (yikes!).

A single source of truth for data ingestion

Data teams building analytics on their warehouse leverage a dedicated tool like RudderStack for behavioral data ingestion. With RudderStack, you can instrument your site with *one single SDK*, capturing event and user identification payloads one time, then syndicating that single version of the truth both to your warehouse as well as other downstream SaaS analytics tools (more on SaaS integrations below).

Data teams rejoice: a single SDK means no more bloated scripts or sequencing or dealing with API changes and broken pipelines. RudderStack's approach also helps when a team wants to switch tools and add a new tool to the stack. You won't need to touch the codebase to delete or add any new SDKs. Instead, you can simply remove the tool or integrate the new tool via the event streaming tool's dashboard.

When it comes to analytics on your warehouse, RudderStack payloads are automatically flattened and loaded into specific tables according to a standardized schema, meaning your analytics team already has a running start when it comes to creating joins and modeling data.

Here are a few specific features of RudderStack's event streaming pipeline that our users find particularly helpful for building rich behavioral analytics:

- Configurable session tracking data in each payload
- Configurable event properties and user traits
- Ad blocker resilience
- Automatic tracking of anonymous users with a unique id
- Multiple methods for identifying users and assigning traits to users and groups, even across domains

3. Prepping the data for analysis: data modeling

Once your behavioral data ingestion pipeline has pulled in events from your websites and apps, the next step is to make sense of all that data.

Data modeling is all about organizing, transforming, and grouping the events to answer the questions teams formerly tried to answer with GA4. For example, say your marketing team wants to understand which channels drive the most content views (i.e., organic search vs paid social). You'll need to count distinct pageviews for each relevant URL, then group them by the referring domain.

It's worth noting that RudderStack's event schemas are specifically designed to make this kind of modeling much easier. In order to perform the above analysis, you'd count pageviews by distinct `anonymousId` (which represents a single user, anonymous or known) to get a unique pageview number, then use the `referrer` value to group pageviews by the site that referred them. To maintain continuity with Google Analytics' default channel grouping, you would assign a channel name to a group of URLs (i.e., Twitter, LinkedIn, and Facebook could be grouped into a channel called 'Social').

Once the data is modeled, it's easier to extract value from it, whether in the form of dashboards, reports, or as a base for predictive or prescriptive analytics.

Many data teams use SQL or Python to model their data, but in the last few years tools like dbt have made the modeling workflow much better for data teams.

What is dbt?

One of the most popular data modeling tools is dbt built by dbtlabs. dbt (data build tool) enables data analytics engineers to transform data in their warehouses by simply writing SQL select statements. dbt takes those SQL codes and runs them against your data warehouse to create tables and views.

dbt enables data engineers to work like software engineers with version control, continuous integration, and testing built-in. Because dbt is SQL-based, it is straightforward to get started with. Plus, dbt is open-source and has a very active community on Slack.

RudderStack offers multiple dbt models that, when run on event data from RudderStack, can scaffold nearly complete data models for key web analytics use cases:

- Sessionization
- Customer journey mapping
- Identity resolution
- Web analytics

Check out the knowledge base article to learn more about the dbt models and how to use them to build analytics

4. Making use of the data: visualization tools

Data modeling is not complete without visualization. Now that you've transformed the data in the warehouse into rich datasets, the next step is to feed it into a visualization tool that will allow you to visually represent those datasets in the form of charts, maps, graphs, or images in order to draw valuable insights from them.

There are a lot of warehouse-focused visualization tools in the market. Some of the popular ones are Tableau, Metabase, Sigma, and Hex. However, before you pick a tool, you should consider things like ease of use, learning curve, support for your choice of data warehouse, level of flexibility, customization options, and if it fits your use case.

Tableau

Tableau is often recognized as the undisputed king of data visualization software and with good cause. Due to its ease of use and capacity to generate interactive visuals well beyond standard BI solutions. It is especially well suited to dealing with large and rapidly changing datasets.

Metabase

Metabase is an open-source business intelligence tool that lets you create charts and dashboards using datasets from your data warehouse. Although SQL is not required to produce visualizations, Metabase does allow SQL for sophisticated customization. Its simplicity and ease of use are the top reasons why many users love it.

Sigma

Sigma is a cloud analytics platform that uses a familiar spreadsheet interface to give business users instant access to explore and get insights from their cloud data warehouse.

Hex

Hex is a modern platform for data science and analytics. From ad-hoc analyses to beautiful dashboards to complex apps, Hex streamlines the entire analytics workflow, so your team can focus on generating insight, driving decisions, and moving things forward.

RICH, CONSISTENT DATA FOR YOUR WHOLE STACK

RudderStack provides the data layer that enables you to get the most out of GA4, unlock deeper analysis in your data warehouse, and send data to every other tool in your stack. Our deep integration library features over 180 tools used by marketing, sales, product, and customer success teams every day. Sign up for free to start improving your analytics today.

RudderStack is the warehouse-first, customer data platform built for developers.

We take a new approach to building and operating your customer data infrastructure, making it easy to collect, unify, transform, and store customer data as well as securely route it to a wide range of marketing, analytics, sales, and product tools.

Over 18,000 sites and apps run RudderStack including Crate & Barrel, Acorns, Hinge, Stripe, Allbirds, and more.

rudderstack.com

@rudderstack