Elevar DataLayer

**Last Updated / Reviewed:** April 11th, 2023

**Execution Time:** *~15 minutes*

**Current Version:** *Version 3.1.10*

**Goal:** To create dataLayer codes inside Shopify to track all ecommerce events, including purchase revenue.

**Ideal Outcome:** Get more than 95% data accuracy into GA4 reporting.

**Configure the Checkout Thank You Page**

If you are on a Shopify Plus store and would like to track the checkout pages, then you'll need to enable "checkout.liquid" first - [View Guide](https://knowledge.getelevar.com/how-to-enable-checkout-liquid-layout)

If you are on a Non-Plus Shopify Store, you will not be able to track the checkout pages - [View Guide](https://knowledge.getelevar.com/can-non-plus-stores-use-gtm-app)

1. Copy the code from the block below:

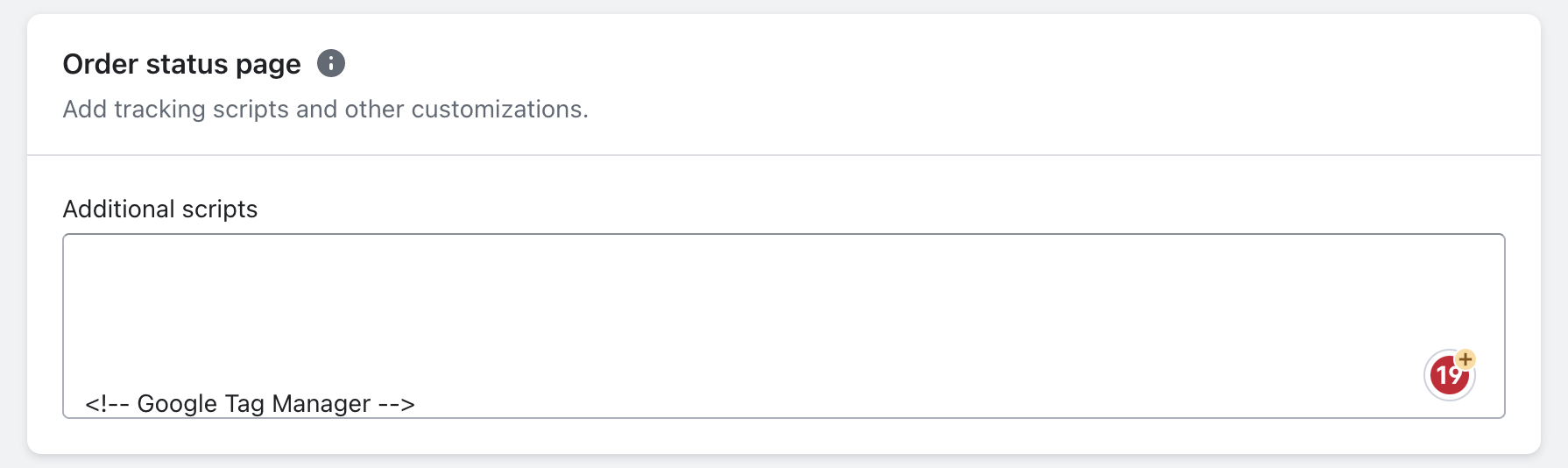
*Please change all GTM container ID in yellow marked*

| <!--  Elevar Data Layer V2  This file is automatically updated and should not be edited directly.  https://knowledge.getelevar.com/how-to-customize-data-layer-version-2  Updated: 2023-04-11 09:11:34+00:00  Version: 3.1.10  -->  <!-- Google Tag Manager -->  <script>  window.dataLayer = window.dataLayer || [];  </script>  <script>  (function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({"gtm.start":  new Date().getTime(),event:"gtm.js"});var f=d.getElementsByTagName(s)[0],  j=d.createElement(s),dl=l!="dataLayer"?"&l="+l:"";j.async=true;j.src=  "https://www.googletagmanager.com/gtm.js?id="+i+dl;f.parentNode.insertBefore(j,f);  })(window,document,"script","dataLayer","GTM-XXXXXXX");  </script>  <!-- End Google Tag Manager -->  <script id="elevar-gtm-suite-config" type="application/json">{"gtm\_id": "GTM-XXXXXXX", "event\_config": {"cart\_reconcile": true, "cart\_view": true, "checkout\_complete": true, "checkout\_step": true, "collection\_view": true, "defers\_collection\_loading": false, "defers\_search\_results\_loading": false, "product\_add\_to\_cart": false, "product\_add\_to\_cart\_ajax": true, "product\_remove\_from\_cart": true, "product\_select": true, "product\_view": true, "search\_results\_view": true, "user": true, "save\_order\_notes": true}, "gtm\_suite\_script": "https://shopify-gtm-suite.getelevar.com/shops/d0b0038eba47fc05bb116d997e301563dd42b6d5/3.1.10/gtm-suite.js", "consent\_enabled": true, "apex\_domain": null}</script>  {% if first\_time\_accessed %}  <script>  (() => {  const init = () => {  window.\_\_ElevarDataLayerQueue = [];  window.\_\_ElevarListenerLoadQueue = [];  if (!window.dataLayer) window.dataLayer = [];  }  init();  window.\_\_ElevarTransformItem = event => {  if (typeof window.ElevarTransformFn === "function") {  try {  const result = window.ElevarTransformFn(event);  if (typeof result === "object" && !Array.isArray(result) && result !== null) {  return result;  } else {  console.error("Elevar Data Layer: `window.ElevarTransformFn` returned a value " + "that wasn't an object, so we've treated things as if this " + "function wasn't defined.");  return event;  }  } catch (error) {  console.error("Elevar Data Layer: `window.ElevarTransformFn` threw an error, so " + "we've treated things as if this function wasn't defined. The " + "exact error is shown below.");  console.error(error);  return event;  }  } else {  return event;  }  }  window.ElevarPushToDataLayer = item => {  const enrichedItem = {  event\_id: window.crypto.randomUUID(),  event\_time: new Date().toISOString(),  ...item  };  const transformedEnrichedItem = window.\_\_ElevarTransformItem ? window.\_\_ElevarTransformItem(enrichedItem) : enrichedItem;  const listenerPayload = {  raw: enrichedItem,  transformed: transformedEnrichedItem  };  const getListenerNotifyEvent = () => {  return new CustomEvent("elevar-listener-notify", {  detail: listenerPayload  });  };  if (transformedEnrichedItem.\_elevar\_internal?.isElevarContextPush) {  window.\_\_ElevarIsContextSet = true;  window.\_\_ElevarDataLayerQueue.unshift(transformedEnrichedItem);  if (window.\_\_ElevarIsListenerListening) {  window.dispatchEvent(getListenerNotifyEvent());  } else {  window.\_\_ElevarListenerLoadQueue.unshift(listenerPayload);  }  } else {  window.\_\_ElevarDataLayerQueue.push(transformedEnrichedItem);  if (window.\_\_ElevarIsListenerListening) {  window.dispatchEvent(getListenerNotifyEvent());  } else {  window.\_\_ElevarListenerLoadQueue.push(listenerPayload);  }  }  if (window.\_\_ElevarIsContextSet) {  while (window.\_\_ElevarDataLayerQueue.length > 0) {  window.dataLayer.push(window.\_\_ElevarDataLayerQueue.shift());  }  }  }  const configElement = document.getElementById("elevar-gtm-suite-config");  if (!configElement) {  console.error("Elevar: DL Config element not found");  return;  }  const config = JSON.parse(configElement.textContent);  const script = document.createElement("script");  script.type = "text/javascript";  script.src = config.gtm\_suite\_script;  script.async = false;  script.defer = true;  script.onerror = () => {  console.error("Elevar: DL JS script failed to load");  };  script.onload = async () => {  if (!window.ElevarGtmSuite) {  console.error("Elevar: `ElevarGtmSuite` is not defined");  return;  }  const cartData = {  attributes: {{- cart.attributes | json -}},  cartTotal: "{{- cart.total\_price | times: 0.01 | json -}}",  currencyCode: {{- cart.currency.iso\_code | json -}},  items: [  {%- for line\_item in cart.items -%}  {  {%- if line\_item.sku != blank -%}  id: {{- line\_item.sku | json -}},  {%- else -%}  id: "{{- line\_item.product\_id | json -}}",  {%- endif -%}  name: {{- line\_item.product.title | json -}},  brand: {{- line\_item.vendor | json -}},  category: {{- line\_item.product.type | json -}},  variant: {{- line\_item.variant.title | json -}},  price: "{{- line\_item.final\_price | times: 0.01 | json -}}",  position: {{- forloop.index -}},  quantity: "{{- line\_item.quantity | json -}}",  productId: "{{- line\_item.product\_id | json -}}",  variantId: "{{- line\_item.variant\_id -}}",  compareAtPrice: "{{- line\_item.variant.compare\_at\_price | times: 0.01 | json -}}",  image: "{{- line\_item.image | image\_url -}}"  },  {%- endfor -%}  ]  }  ;  await window.ElevarGtmSuite.handlers.cartAttributesReconcile(  cartData,  config.event\_config.save\_order\_notes,  config.consent\_enabled,  config.apex\_domain  );  if (config.event\_config.user) {  const data = {  {%- if checkout -%}  cartTotal: "{{- checkout.total\_price | times: 0.01 | json -}}",  currencyCode: {{- checkout.currency | json -}},  {%- else -%}  cartTotal: "{{- cart.total\_price | times: 0.01 | json -}}",  currencyCode: {{- cart.currency.iso\_code | json -}},  {%- endif -%}  {%- if customer -%}  customer: {  id: "{{- customer.id | json -}}",  email: {{- customer.email | json -}},  firstName: {{- customer.first\_name | json -}},  lastName: {{- customer.last\_name | json -}},  phone: {{- customer.phone | json -}},  city: {{- customer.default\_address.city | json -}},  zip: {{- customer.default\_address.zip | json -}},  address1: {{- customer.default\_address.address1 | json -}},  address2: {{- customer.default\_address.address2 | json -}},  country: {{- customer.default\_address.country | json -}},  province: {{- customer.default\_address.province | json -}},  provinceCode: {{- customer.default\_address.province\_code | json -}},  orderCount: "{{- customer.orders\_count | json -}}",  totalSpent: "{{- customer.total\_spent | times: 0.01 | json -}}",  tags: {{- customer.tags | join: ', ' | json -}}  }  {%- endif -%}  };  window.ElevarGtmSuite.handlers.user(data);  }  {%- if checkout -%}  {%- if checkout.attributes["\_elevar\_visitor\_info"] == blank -%}  if (config.event\_config.save\_order\_notes) {  const data = {%- if checkout -%}  {  checkoutId: {{- checkout.id | json -}}  }  {%- endif -%}  ;  window.ElevarGtmSuite.handlers.storeOrderNotesOffsite(data);  }  {%- endif -%}  if (config.event\_config.checkout\_complete) {  const data = {%- if checkout -%}  {  currencyCode: {{- checkout.currency | json -}},  actionField: {  {%- if checkout.order\_id -%}  id: {{- checkout.order\_id | json -}},  {%- else -%}  id: {{- checkout.id | json -}},  {%- endif -%}  {%- if checkout.order\_name -%}  order\_name: {{- checkout.order\_name | json -}},  {%- endif -%}  affiliation: {{- shop.name | json -}},  revenue: "{{- checkout.total\_price | times: 0.01 | json -}}",  tax: "{{- checkout.tax\_price | times: 0.01 | json -}}",  shipping: "{{- checkout.shipping\_price | times: 0.01 | json -}}",  {% if checkout.discount\_applications %}  coupon: {{ checkout.discount\_applications[0].title | json }},  {% endif %}  {%- if order.subtotal\_price -%}  subTotal: "{{- order.subtotal\_price | times: 0.01 | json -}}",  {%- elsif checkout.subtotal\_price -%}  subTotal: "{{- checkout.subtotal\_price | times: 0.01 | json -}}",  {%- endif -%}  productSubTotal: "{{- checkout.line\_items\_subtotal\_price | times: 0.01 | json -}}",  discountAmount: "{{- checkout.discounts\_amount | times: 0.01 | json -}}"  },  {%- if checkout.customer -%}  customer: {  id: "{{- checkout.customer.id | json -}}",  email: {{- checkout.email | json -}},  firstName: {{- checkout.billing\_address.first\_name | json -}},  lastName: {{- checkout.billing\_address.last\_name | json -}},  {%- if checkout.customer.phone -%}  phone: {{- checkout.customer.phone | json -}},  {%- elsif checkout.billing\_address.phone -%}  phone: {{- checkout.billing\_address.phone | json -}},  {%- else -%}  phone: {{- checkout.shipping\_address.phone | json -}},  {%- endif -%}  city: {{- checkout.billing\_address.city | json -}},  zip: {{- checkout.billing\_address.zip | json -}},  address1: {{- checkout.billing\_address.address1 | json -}},  address2: {{- checkout.billing\_address.address2 | json -}},  country: {{- checkout.billing\_address.country | json -}},  province: {{- checkout.billing\_address.province | json -}},  provinceCode: {{- checkout.billing\_address.province\_code | json -}},  orderCount: "{{- checkout.customer.orders\_count | json -}}",  totalSpent: "{{- checkout.customer.total\_spent | times: 0.01 | json -}}",  tags: {{- checkout.customer.tags | json -}}  },  {%- endif -%}  items: [  {%- for line\_item in checkout.line\_items -%}  {  id: {{- line\_item.sku | json -}},  name: {{- line\_item.product.title | json -}},  brand: {{- line\_item.vendor | json -}},  category: {{- line\_item.product.type | json -}},  variant: {{- line\_item.variant.title | json -}},  price: "{{- line\_item.final\_price | times: 0.01 | json -}}",  quantity: "{{- line\_item.quantity | json -}}",  productId: "{{- line\_item.product\_id | json -}}",  variantId: "{{- line\_item.variant\_id -}}",  image: "{{- line\_item.image | image\_url -}}"  },  {%- endfor -%}  ],  landingSite: {{- checkout.landing\_site | json -}}  }  {%- endif -%}  ;  window.ElevarGtmSuite.handlers.checkoutComplete(data);  }  {%- endif -%}  };  document.head.appendChild(script);  })();  </script>  {% endif %}  <!-- Google Tag Manager (noscript) -->  <noscript>  <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-XXXXXXX" height="0" width="0" style="display:none;visibility:hidden"></iframe>  </noscript>  <!-- End Google Tag Manager (noscript) --> |
| --- |

2. Open your "Checkout Settings" admin page (**settings > Checkout**).

3. Scroll down to the "**Order status page**" section of the page.

4. Paste the code from above into the end of the "**Additional scripts**" input box:



5. Click "Save".

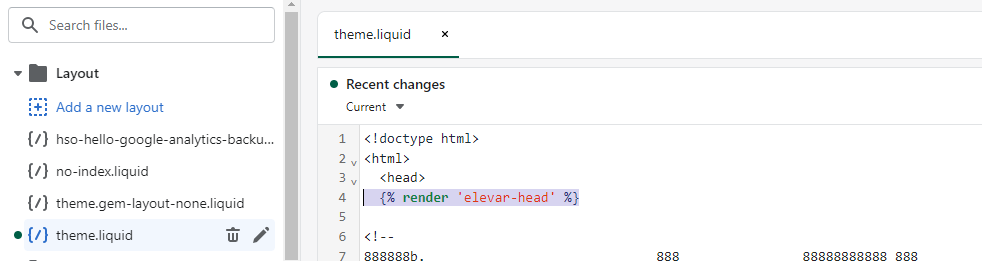
**Step-01: Install Data Layer 1st call code on Shopify**

**Login into Shopify admin panel:**

Online Store----> Themes ----> Actions ----> Edit Code ----> Layout ----> theme.liquid (File) ---->

> Put the Data Layer call code directly below the opening <head> tag

**{% render 'elevar-head' %}**



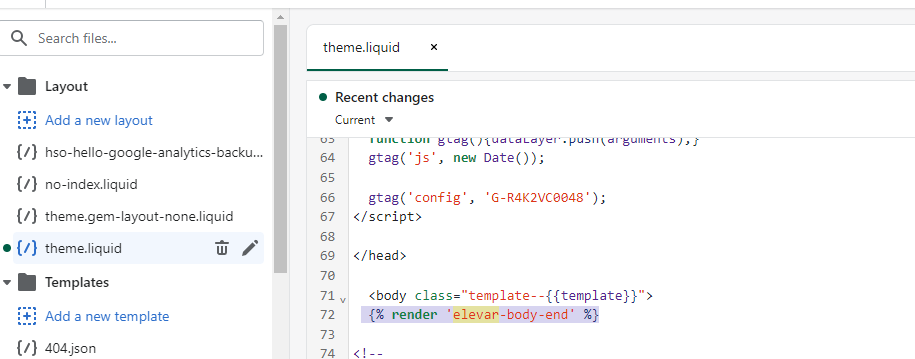
**Step-02: Install Data Layer call 2nd code on Shopify**

**Login into Shopify admin panel:**

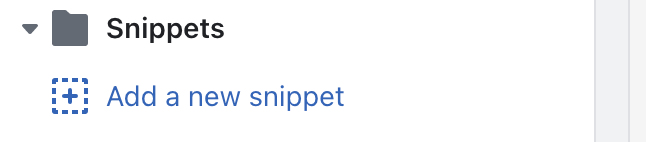
Online Store----> Themes ----> Actions ----> Edit Code ----> Layout ----> theme.liquid (File) ---->

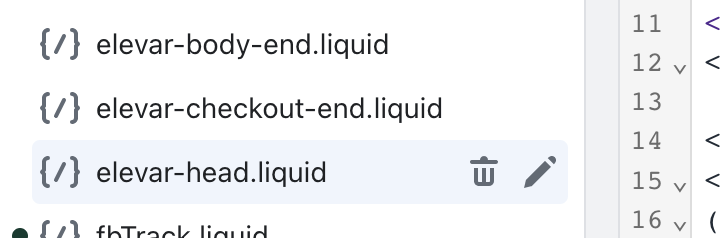
> Put the Data Layer call code directly below the opening fast <body> tag

**{% render 'elevar-body-end' %}**



**Create Snippet Data Layer**





**Step-3: Create the dataLayer-allPages snippet ( use exact naming and casing) & copy and paste the below code**

Go to Online Store ----> Themes ----> Snippets ----> Add a new snippet

Name: **elevar-head**

| <!--  Elevar Data Layer V2  This file is automatically updated and should not be edited directly.  https://knowledge.getelevar.com/how-to-customize-data-layer-version-2  Updated: 2023-04-11 09:36:56+00:00  Version: 3.1.10  -->  <!-- Google Tag Manager -->  <script>  window.dataLayer = window.dataLayer || [];  </script>  <script>  (function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({"gtm.start":  new Date().getTime(),event:"gtm.js"});var f=d.getElementsByTagName(s)[0],  j=d.createElement(s),dl=l!="dataLayer"?"&l="+l:"";j.async=true;j.src=  "https://www.googletagmanager.com/gtm.js?id="+i+dl;f.parentNode.insertBefore(j,f);  })(window,document,"script","dataLayer","GTM-XXXXXX");  </script>  <!-- End Google Tag Manager -->  <script id="elevar-gtm-suite-config" type="application/json">{"gtm\_id": "GTM-XXXXX", "event\_config": {"cart\_reconcile": true, "cart\_view": true, "checkout\_complete": true, "checkout\_step": true, "collection\_view": true, "defers\_collection\_loading": false, "defers\_search\_results\_loading": false, "product\_add\_to\_cart": false, "product\_add\_to\_cart\_ajax": true, "product\_remove\_from\_cart": true, "product\_select": true, "product\_view": true, "search\_results\_view": true, "user": true, "save\_order\_notes": true}, "gtm\_suite\_script": "https://shopify-gtm-suite.getelevar.com/shops/d0b0038eba47fc05bb116d997e301563dd42b6d5/3.1.10/gtm-suite.js", "consent\_enabled": true, "apex\_domain": null}</script> |
| --- |

**Step-4: Create the dataLayer-allPages snippet ( use exact naming and casing) & copy and paste the below code**

Go to Online Store ----> Themes ----> Snippets ----> Add a new snippet

Name: **elevar-checkout-end**

| <script>  (() => {  const init = () => {  window.\_\_ElevarDataLayerQueue = [];  window.\_\_ElevarListenerLoadQueue = [];  if (!window.dataLayer) window.dataLayer = [];  }  init();  window.\_\_ElevarTransformItem = event => {  if (typeof window.ElevarTransformFn === "function") {  try {  const result = window.ElevarTransformFn(event);  if (typeof result === "object" && !Array.isArray(result) && result !== null) {  return result;  } else {  console.error("Elevar Data Layer: `window.ElevarTransformFn` returned a value " + "that wasn't an object, so we've treated things as if this " + "function wasn't defined.");  return event;  }  } catch (error) {  console.error("Elevar Data Layer: `window.ElevarTransformFn` threw an error, so " + "we've treated things as if this function wasn't defined. The " + "exact error is shown below.");  console.error(error);  return event;  }  } else {  return event;  }  }  window.ElevarPushToDataLayer = item => {  const enrichedItem = {  event\_id: window.crypto.randomUUID(),  event\_time: new Date().toISOString(),  ...item  };  const transformedEnrichedItem = window.\_\_ElevarTransformItem ? window.\_\_ElevarTransformItem(enrichedItem) : enrichedItem;  const listenerPayload = {  raw: enrichedItem,  transformed: transformedEnrichedItem  };  const getListenerNotifyEvent = () => {  return new CustomEvent("elevar-listener-notify", {  detail: listenerPayload  });  };  if (transformedEnrichedItem.\_elevar\_internal?.isElevarContextPush) {  window.\_\_ElevarIsContextSet = true;  window.\_\_ElevarDataLayerQueue.unshift(transformedEnrichedItem);  if (window.\_\_ElevarIsListenerListening) {  window.dispatchEvent(getListenerNotifyEvent());  } else {  window.\_\_ElevarListenerLoadQueue.unshift(listenerPayload);  }  } else {  window.\_\_ElevarDataLayerQueue.push(transformedEnrichedItem);  if (window.\_\_ElevarIsListenerListening) {  window.dispatchEvent(getListenerNotifyEvent());  } else {  window.\_\_ElevarListenerLoadQueue.push(listenerPayload);  }  }  if (window.\_\_ElevarIsContextSet) {  while (window.\_\_ElevarDataLayerQueue.length > 0) {  window.dataLayer.push(window.\_\_ElevarDataLayerQueue.shift());  }  }  }  const configElement = document.getElementById("elevar-gtm-suite-config");  if (!configElement) {  console.error("Elevar: DL Config element not found");  return;  }  const config = JSON.parse(configElement.textContent);  const script = document.createElement("script");  script.type = "text/javascript";  script.src = config.gtm\_suite\_script;  script.async = false;  script.defer = true;  script.onerror = () => {  console.error("Elevar: DL JS script failed to load");  };  script.onload = async () => {  if (!window.ElevarGtmSuite) {  console.error("Elevar: `ElevarGtmSuite` is not defined");  return;  }  const cartData = {  attributes: {{- cart.attributes | json -}},  cartTotal: "{{- cart.total\_price | times: 0.01 | json -}}",  currencyCode: {{- cart.currency.iso\_code | json -}},  items: [  {%- for line\_item in cart.items -%}  {  {%- if line\_item.sku != blank -%}  id: {{- line\_item.sku | json -}},  {%- else -%}  id: "{{- line\_item.product\_id | json -}}",  {%- endif -%}  name: {{- line\_item.product.title | json -}},  brand: {{- line\_item.vendor | json -}},  category: {{- line\_item.product.type | json -}},  variant: {{- line\_item.variant.title | json -}},  price: "{{- line\_item.final\_price | times: 0.01 | json -}}",  position: {{- forloop.index -}},  quantity: "{{- line\_item.quantity | json -}}",  productId: "{{- line\_item.product\_id | json -}}",  variantId: "{{- line\_item.variant\_id -}}",  compareAtPrice: "{{- line\_item.variant.compare\_at\_price | times: 0.01 | json -}}",  image: "{{- line\_item.image | image\_url -}}"  },  {%- endfor -%}  ]  }  ;  await window.ElevarGtmSuite.handlers.cartAttributesReconcile(  cartData,  config.event\_config.save\_order\_notes,  config.consent\_enabled,  config.apex\_domain  );  if (config.event\_config.user) {  const data = {  {%- if checkout -%}  cartTotal: "{{- checkout.total\_price | times: 0.01 | json -}}",  currencyCode: {{- checkout.currency | json -}},  {%- else -%}  cartTotal: "{{- cart.total\_price | times: 0.01 | json -}}",  currencyCode: {{- cart.currency.iso\_code | json -}},  {%- endif -%}  {%- if customer -%}  customer: {  id: "{{- customer.id | json -}}",  email: {{- customer.email | json -}},  firstName: {{- customer.first\_name | json -}},  lastName: {{- customer.last\_name | json -}},  phone: {{- customer.phone | json -}},  city: {{- customer.default\_address.city | json -}},  zip: {{- customer.default\_address.zip | json -}},  address1: {{- customer.default\_address.address1 | json -}},  address2: {{- customer.default\_address.address2 | json -}},  country: {{- customer.default\_address.country | json -}},  province: {{- customer.default\_address.province | json -}},  provinceCode: {{- customer.default\_address.province\_code | json -}},  orderCount: "{{- customer.orders\_count | json -}}",  totalSpent: "{{- customer.total\_spent | times: 0.01 | json -}}",  tags: {{- customer.tags | join: ', ' | json -}}  }  {%- endif -%}  };  window.ElevarGtmSuite.handlers.user(data);  }  {%- if checkout -%}  const step = Shopify?.Checkout?.step;  const page = Shopify?.Checkout?.page;  const isOnBeginCheckout = step === "contact\_information";  const isOnThankYouPage = page?.includes("thank\_you");  {%- if checkout.attributes["\_elevar\_visitor\_info"] == blank -%}  if (config.event\_config.save\_order\_notes && Shopify?.Checkout?.token) {  if (isOnBeginCheckout) {  const data = {%- if checkout -%}  {  token: Shopify?.Checkout?.token,  currency: {{- checkout.currency | json -}},  {%- if customer -%}  customer: { email: {{- customer.email | json -}} },  {%- else -%}  customer: {},  {%- endif -%}  line\_items: [  {%- for line\_item in checkout.line\_items -%}  {  id: {{- line\_item.id | json -}},  quantity: {{- line\_item.quantity | json -}},  title: {{- line\_item.title | json -}},  variant: {  id: {{- line\_item.variant.id | json -}},  price: {{- line\_item.variant.price | json -}},  sku: {{- line\_item.variant.sku | json -}},  title: {{- line\_item.variant.title | json -}},  product: {  id: {{- line\_item.variant.product.id | json -}},  title: {{- line\_item.variant.product.title | json -}},  vendor: {{- line\_item.variant.product.vendor | json -}}  }  }  },  {%- endfor -%}  ],  {%- if checkout.shipping\_address -%}  shipping\_address: {  city: {{- checkout.shipping\_address.city | json -}},  country: {{- checkout.shipping\_address.country | json -}},  country\_code: {{- checkout.shipping\_address.country\_code | json -}},  phone: {{- checkout.shipping\_address.phone | json -}},  province: {{- checkout.shipping\_address.province | json -}},  province\_code: {{- checkout.shipping\_address.province\_code | json -}}  },  {%- else -%}  shipping\_address: {},  {%- endif -%}  shipping\_price: {{- checkout.shipping\_price | json -}},  subtotal\_price: {{- checkout.order.subtotal\_price | json -}},  total\_price: {{- checkout.total\_price | json -}},  total\_tax: {{- checkout.tax\_price | json -}}  }  {%- endif -%}  ;  window.ElevarGtmSuite.handlers.storeOrderNotesOffsite(data);  } else {  const data = {%- if checkout -%}  {  checkoutId: {{- checkout.id | json -}}  }  {%- endif -%}  ;  window.ElevarGtmSuite.handlers.storeOrderNotesOffsite(data);  }  }  {%- endif -%}  {% if request.path contains "thank\_you" %}  if (config.event\_config.checkout\_complete && isOnThankYouPage) {  const data = {%- if checkout -%}  {  currencyCode: {{- checkout.currency | json -}},  actionField: {  {%- if checkout.order\_id -%}  id: {{- checkout.order\_id | json -}},  {%- else -%}  id: {{- checkout.id | json -}},  {%- endif -%}  {%- if checkout.order\_name -%}  order\_name: {{- checkout.order\_name | json -}},  {%- endif -%}  affiliation: {{- shop.name | json -}},  revenue: "{{- checkout.total\_price | times: 0.01 | json -}}",  tax: "{{- checkout.tax\_price | times: 0.01 | json -}}",  shipping: "{{- checkout.shipping\_price | times: 0.01 | json -}}",  {% if checkout.discount\_applications %}  coupon: {{ checkout.discount\_applications[0].title | json }},  {% endif %}  {%- if order.subtotal\_price -%}  subTotal: "{{- order.subtotal\_price | times: 0.01 | json -}}",  {%- elsif checkout.subtotal\_price -%}  subTotal: "{{- checkout.subtotal\_price | times: 0.01 | json -}}",  {%- endif -%}  productSubTotal: "{{- checkout.line\_items\_subtotal\_price | times: 0.01 | json -}}",  discountAmount: "{{- checkout.discounts\_amount | times: 0.01 | json -}}"  },  {%- if checkout.customer -%}  customer: {  id: "{{- checkout.customer.id | json -}}",  email: {{- checkout.email | json -}},  firstName: {{- checkout.billing\_address.first\_name | json -}},  lastName: {{- checkout.billing\_address.last\_name | json -}},  {%- if checkout.customer.phone -%}  phone: {{- checkout.customer.phone | json -}},  {%- elsif checkout.billing\_address.phone -%}  phone: {{- checkout.billing\_address.phone | json -}},  {%- else -%}  phone: {{- checkout.shipping\_address.phone | json -}},  {%- endif -%}  city: {{- checkout.billing\_address.city | json -}},  zip: {{- checkout.billing\_address.zip | json -}},  address1: {{- checkout.billing\_address.address1 | json -}},  address2: {{- checkout.billing\_address.address2 | json -}},  country: {{- checkout.billing\_address.country | json -}},  province: {{- checkout.billing\_address.province | json -}},  provinceCode: {{- checkout.billing\_address.province\_code | json -}},  orderCount: "{{- checkout.customer.orders\_count | json -}}",  totalSpent: "{{- checkout.customer.total\_spent | times: 0.01 | json -}}",  tags: {{- checkout.customer.tags | json -}}  },  {%- endif -%}  items: [  {%- for line\_item in checkout.line\_items -%}  {  id: {{- line\_item.sku | json -}},  name: {{- line\_item.product.title | json -}},  brand: {{- line\_item.vendor | json -}},  category: {{- line\_item.product.type | json -}},  variant: {{- line\_item.variant.title | json -}},  price: "{{- line\_item.final\_price | times: 0.01 | json -}}",  quantity: "{{- line\_item.quantity | json -}}",  productId: "{{- line\_item.product\_id | json -}}",  variantId: "{{- line\_item.variant\_id -}}",  image: "{{- line\_item.image | image\_url -}}"  },  {%- endfor -%}  ],  landingSite: {{- checkout.landing\_site | json -}}  }  {%- endif -%}  ;  window.ElevarGtmSuite.handlers.checkoutComplete(data);  }  {%- endif -%}  if (Shopify?.Checkout && !isOnThankYouPage) {  const data = {%- if checkout -%}  {  currencyCode: {{- checkout.currency | json -}},  items: [  {%- for line\_item in checkout.line\_items -%}  {  id: {{- line\_item.sku | json -}},  name: {{- line\_item.product.title | json -}},  brand: {{- line\_item.vendor | json -}},  category: {{- line\_item.product.type | json -}},  variant: {{- line\_item.variant.title | json -}},  price: "{{- line\_item.final\_price | times: 0.01 | json -}}",  quantity: "{{- line\_item.quantity | json -}}",  productId: "{{- line\_item.product\_id | json -}}",  variantId: "{{- line\_item.variant\_id -}}",  compareAtPrice: "{{- line\_item.variant.compare\_at\_price | times: 0.01 | json -}}",  image: "{{- line\_item.image | image\_url -}}"  },  {%- endfor -%}  ]  }  {%- endif -%}  ;  if (config.event\_config.cart\_reconcile) {  window.ElevarGtmSuite.handlers.cartItemsReconcile(data);  }  if (config.event\_config.checkout\_step) {  window.ElevarGtmSuite.handlers.checkoutStep(data);  }  }  {%- endif -%}  };  document.head.appendChild(script);  })();  </script>  <!-- Google Tag Manager (noscript) -->  <noscript>  <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-XXXXXX" height="0" width="0" style="display:none;visibility:hidden"></iframe>  </noscript>  <!-- End Google Tag Manager (noscript) --> |
| --- |

**Step-05: Create the dataLayer-allPages snippet ( use exact naming and casing) & copy and paste the below code**

Go to Online Store ----> Themes ----> Snippets ----> Add a new snippet

Name: **elevar-body-end**

| <script>  (() => {  const init = () => {  window.\_\_ElevarDataLayerQueue = [];  window.\_\_ElevarListenerLoadQueue = [];  if (!window.dataLayer) window.dataLayer = [];  }  init();  window.\_\_ElevarTransformItem = event => {  if (typeof window.ElevarTransformFn === "function") {  try {  const result = window.ElevarTransformFn(event);  if (typeof result === "object" && !Array.isArray(result) && result !== null) {  return result;  } else {  console.error("Elevar Data Layer: `window.ElevarTransformFn` returned a value " + "that wasn't an object, so we've treated things as if this " + "function wasn't defined.");  return event;  }  } catch (error) {  console.error("Elevar Data Layer: `window.ElevarTransformFn` threw an error, so " + "we've treated things as if this function wasn't defined. The " + "exact error is shown below.");  console.error(error);  return event;  }  } else {  return event;  }  }  window.ElevarPushToDataLayer = item => {  const enrichedItem = {  event\_id: window.crypto.randomUUID(),  event\_time: new Date().toISOString(),  ...item  };  const transformedEnrichedItem = window.\_\_ElevarTransformItem ? window.\_\_ElevarTransformItem(enrichedItem) : enrichedItem;  const listenerPayload = {  raw: enrichedItem,  transformed: transformedEnrichedItem  };  const getListenerNotifyEvent = () => {  return new CustomEvent("elevar-listener-notify", {  detail: listenerPayload  });  };  if (transformedEnrichedItem.\_elevar\_internal?.isElevarContextPush) {  window.\_\_ElevarIsContextSet = true;  window.\_\_ElevarDataLayerQueue.unshift(transformedEnrichedItem);  if (window.\_\_ElevarIsListenerListening) {  window.dispatchEvent(getListenerNotifyEvent());  } else {  window.\_\_ElevarListenerLoadQueue.unshift(listenerPayload);  }  } else {  window.\_\_ElevarDataLayerQueue.push(transformedEnrichedItem);  if (window.\_\_ElevarIsListenerListening) {  window.dispatchEvent(getListenerNotifyEvent());  } else {  window.\_\_ElevarListenerLoadQueue.push(listenerPayload);  }  }  if (window.\_\_ElevarIsContextSet) {  while (window.\_\_ElevarDataLayerQueue.length > 0) {  window.dataLayer.push(window.\_\_ElevarDataLayerQueue.shift());  }  }  }  const configElement = document.getElementById("elevar-gtm-suite-config");  if (!configElement) {  console.error("Elevar: DL Config element not found");  return;  }  let initialProductAnchorElements = null;  const setElements = () => {  initialProductAnchorElements = document.querySelectorAll(  `a[href\*="/products/"]:not(a[href\*="/collections/products/"]:not(a[href\*="/collections/products/products/"]))`  );  };  if (document.readyState === "loading") {  document.addEventListener("readystatechange", setElements);  } else {  setElements();  }  const config = JSON.parse(configElement.textContent);  const script = document.createElement("script");  script.type = "text/javascript";  script.src = config.gtm\_suite\_script;  script.async = false;  script.defer = true;  script.onerror = () => {  console.error("Elevar: DL JS script failed to load");  };  script.onload = async () => {  if (!window.ElevarGtmSuite) {  console.error("Elevar: `ElevarGtmSuite` is not defined");  return;  }  const cartData = {  attributes: {{- cart.attributes | json -}},  cartTotal: "{{- cart.total\_price | times: 0.01 | json -}}",  currencyCode: {{- cart.currency.iso\_code | json -}},  items: [  {%- for line\_item in cart.items -%}  {  {%- if line\_item.sku != blank -%}  id: {{- line\_item.sku | json -}},  {%- else -%}  id: "{{- line\_item.product\_id | json -}}",  {%- endif -%}  name: {{- line\_item.product.title | json -}},  brand: {{- line\_item.vendor | json -}},  category: {{- line\_item.product.type | json -}},  variant: {{- line\_item.variant.title | json -}},  price: "{{- line\_item.final\_price | times: 0.01 | json -}}",  position: {{- forloop.index -}},  quantity: "{{- line\_item.quantity | json -}}",  productId: "{{- line\_item.product\_id | json -}}",  variantId: "{{- line\_item.variant\_id -}}",  compareAtPrice: "{{- line\_item.variant.compare\_at\_price | times: 0.01 | json -}}",  image: "{{- line\_item.image | image\_url -}}"  },  {%- endfor -%}  ]  }  ;  await window.ElevarGtmSuite.handlers.cartAttributesReconcile(  cartData,  config.event\_config.save\_order\_notes,  config.consent\_enabled,  config.apex\_domain  );  if (config.event\_config.user) {  const data = {  {%- if checkout -%}  cartTotal: "{{- checkout.total\_price | times: 0.01 | json -}}",  currencyCode: {{- checkout.currency | json -}},  {%- else -%}  cartTotal: "{{- cart.total\_price | times: 0.01 | json -}}",  currencyCode: {{- cart.currency.iso\_code | json -}},  {%- endif -%}  {%- if customer -%}  customer: {  id: "{{- customer.id | json -}}",  email: {{- customer.email | json -}},  firstName: {{- customer.first\_name | json -}},  lastName: {{- customer.last\_name | json -}},  phone: {{- customer.phone | json -}},  city: {{- customer.default\_address.city | json -}},  zip: {{- customer.default\_address.zip | json -}},  address1: {{- customer.default\_address.address1 | json -}},  address2: {{- customer.default\_address.address2 | json -}},  country: {{- customer.default\_address.country | json -}},  province: {{- customer.default\_address.province | json -}},  provinceCode: {{- customer.default\_address.province\_code | json -}},  orderCount: "{{- customer.orders\_count | json -}}",  totalSpent: "{{- customer.total\_spent | times: 0.01 | json -}}",  tags: {{- customer.tags | join: ', ' | json -}}  }  {%- endif -%}  };  window.ElevarGtmSuite.handlers.user(data);  }  if (config.event\_config.product\_add\_to\_cart\_ajax) {  window.ElevarGtmSuite.handlers.productAddToCartAjax(  config.event\_config.save\_order\_notes,  config.consent\_enabled,  config.apex\_domain  );  }  if (config.event\_config.cart\_reconcile) {  window.ElevarGtmSuite.handlers.cartItemsReconcile(cartData);  }  {%- if request.page\_type == "cart" -%}  if (config.event\_config.cart\_view) {  window.ElevarGtmSuite.handlers.cartView(cartData);  }  if (config.event\_config.product\_remove\_from\_cart) {  window.ElevarGtmSuite.handlers.productRemoveFromCart(cartData);  }  {%- endif -%}  {%- if collection and template.name == "collection" -%}  const data = {%- if collection and template.name == "collection" -%}  {  currencyCode: {{- cart.currency.iso\_code | json -}},  items: [  {%- assign itemIndex = 1 -%}  {%- for product in collection.products -%}  {% assign default\_variant = product.selected\_or\_first\_available\_variant %}  {  {%- if default\_variant.sku != blank -%}  id: {{- default\_variant.sku | json -}},  {%- else -%}  id: "{{- default\_variant.id | json -}}",  {%- endif -%}  name: {{- product.title | json -}},  brand: {{- product.vendor | json -}},  category: {{- product.type | json -}},  variant: {{- default\_variant.title | json -}},  price: "{{- product.price | times: 0.01 | json -}}",  position: {{- itemIndex -}},  productId: "{{- product.id | json -}}",  variantId: "{{- default\_variant.id | json -}}",  handle: {{- product.handle | json -}}  },  {%- assign itemIndex = itemIndex | plus: 1 -%}  {%- endfor -%}  ]  }  {%- endif -%};  await window.ElevarGtmSuite.handlers.collectionView(  data,  config.event\_config.collection\_view,  config.event\_config.defers\_collection\_loading,  initialProductAnchorElements  );  if (config.event\_config.product\_select) {  window.ElevarGtmSuite.handlers.productSelect(data);  }  {%- endif -%}  {%- if search.performed -%}  const data = {%- if search.performed -%}  {  currencyCode: {{- cart.currency.iso\_code | json -}},  items: [  {%- assign itemIndex = 1 -%}  {%- for product in search.results -%}  {%- if product.object\_type == 'product' -%}  {%- for variant in product.variants -%}  {  {%- if variant.sku != blank -%}  id: {{- variant.sku | json -}},  {%- else -%}  id: "{{- variant.id | json -}}",  {%- endif -%}  name: {{- product.title | json -}},  brand: {{- product.vendor | json -}},  category: {{- product.type | json -}},  variant: {{- variant.title | json -}},  price: "{{- product.price | times: 0.01 | json -}}",  position: {{- itemIndex -}},  productId: "{{- product.id | json -}}",  variantId: "{{- variant.id | json -}}",  handle: {{- product.handle | json -}}  },  {%- endfor -%}  {%- assign itemIndex = itemIndex | plus: 1 -%}  {%- endif -%}  {%- endfor -%}  ]  }  {%- endif -%}  ;  await window.ElevarGtmSuite.handlers.searchResultsView(  data,  config.event\_config.search\_results\_view,  config.event\_config.defers\_search\_results\_loading,  initialProductAnchorElements  );  if (config.event\_config.product\_select) {  window.ElevarGtmSuite.handlers.productSelect(data);  }  {%- endif -%}  {%- if product -%}  const data = {%- if product -%}  {  attributes: {{- cart.attributes | json -}},  currencyCode: {{- cart.currency.iso\_code | json -}},  {% assign default\_variant = product.selected\_or\_first\_available\_variant %}  {%- if default\_variant != blank -%}  defaultVariant: {  {%- if default\_variant.sku != blank -%}  id: {{- default\_variant.sku | json -}},  {%- else -%}  id: "{{- default\_variant.id | json -}}",  {%- endif -%}  name: {{- product.title | json -}},  brand: {{- product.vendor | json -}},  category: {{- product.type | json -}},  variant: {{- default\_variant.title | json -}},  price: "{{- default\_variant.price | times: 0.01 | json -}}",  productId: "{{- product.id | json -}}",  variantId: "{{- default\_variant.id | json -}}",  compareAtPrice: "{{- default\_variant.compare\_at\_price | times: 0.01 | json -}}",  image: {{- product.featured\_image | json -}},  inventory: "{{- default\_variant.inventory\_quantity -}}"  },  {%- else -%}  defaultVariant: null,  {%- endif -%}  items: [  {%- for variant in product.variants -%}  {  {%- if variant.sku != blank -%}  id: {{- variant.sku | json -}},  {%- else -%}  id: "{{- variant.id | json -}}",  {%- endif -%}  name: {{- product.title | json -}},  brand: {{- product.vendor | json -}},  category: {{- product.type | json -}},  variant: {{- variant.title | json -}},  price: "{{- variant.price | times: 0.01 | json -}}",  productId: "{{- product.id | json -}}",  variantId: "{{- variant.id | json -}}",  compareAtPrice: "{{- variant.compare\_at\_price | times: 0.01 | json -}}",  image: {{- product.featured\_image | json -}},  inventory: "{{- variant.inventory\_quantity -}}"  },  {%- endfor -%}  ]  }  {%- endif -%};  if (config.event\_config.product\_view) {  window.ElevarGtmSuite.handlers.productView(data);  }  if (config.event\_config.product\_add\_to\_cart) {  window.ElevarGtmSuite.handlers.productAddToCartForm(  data,  config.event\_config.save\_order\_notes,  config.consent\_enabled,  config.apex\_domain  );  }  {%- endif -%}  };  document.head.appendChild(script);  })();  </script>  <!-- Google Tag Manager (noscript) -->  <noscript>  <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-XXXXXX" height="0" width="0" style="display:none;visibility:hidden"></iframe>  </noscript>  <!-- End Google Tag Manager (noscript) --> |
| --- |

**Source – Data Layer Listener**

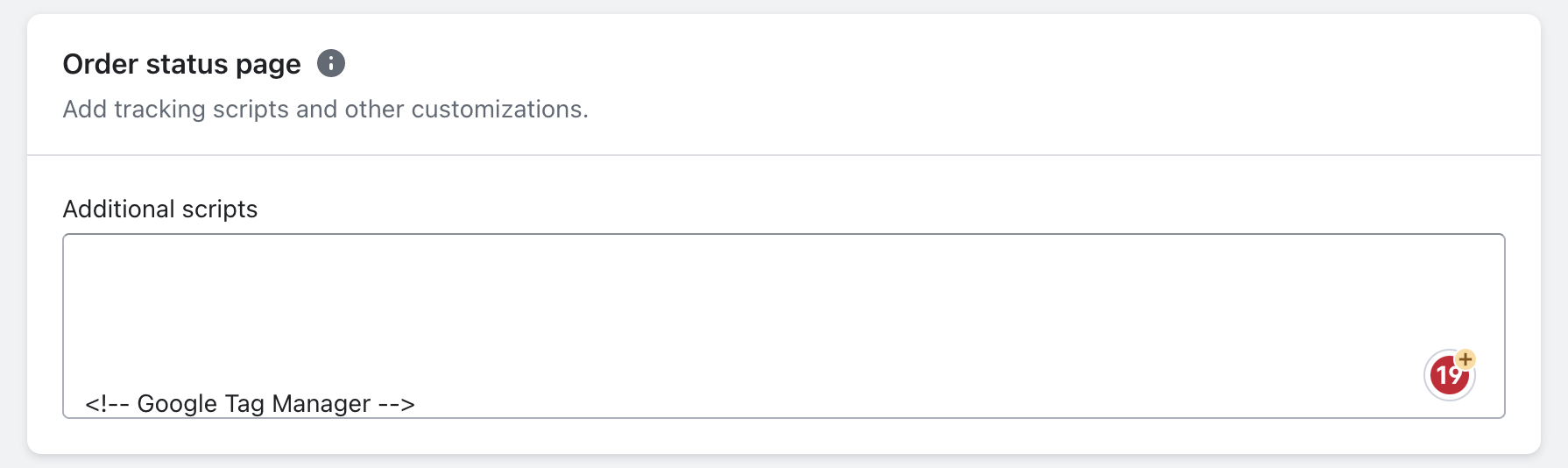
To ensure transaction data is available to GTM, you'll need to manually add an additional script to the "Thank You" page via your Shopify admin. This is in addition to the code copied from the data layer settings.

1. Copy the code from the block below:

2. Open your "Checkout Settings" admin page (by clicking here).

3. Scroll down to the "Order status page" section of the page.

4. Paste the code from above into the end of the "Additional scripts" input box (right below where you pasted the Data Layer code):



5. Click "Save".

**Step-06: Create the dataLayer-allPages snippet ( use exact naming and casing) & copy and paste the below code**

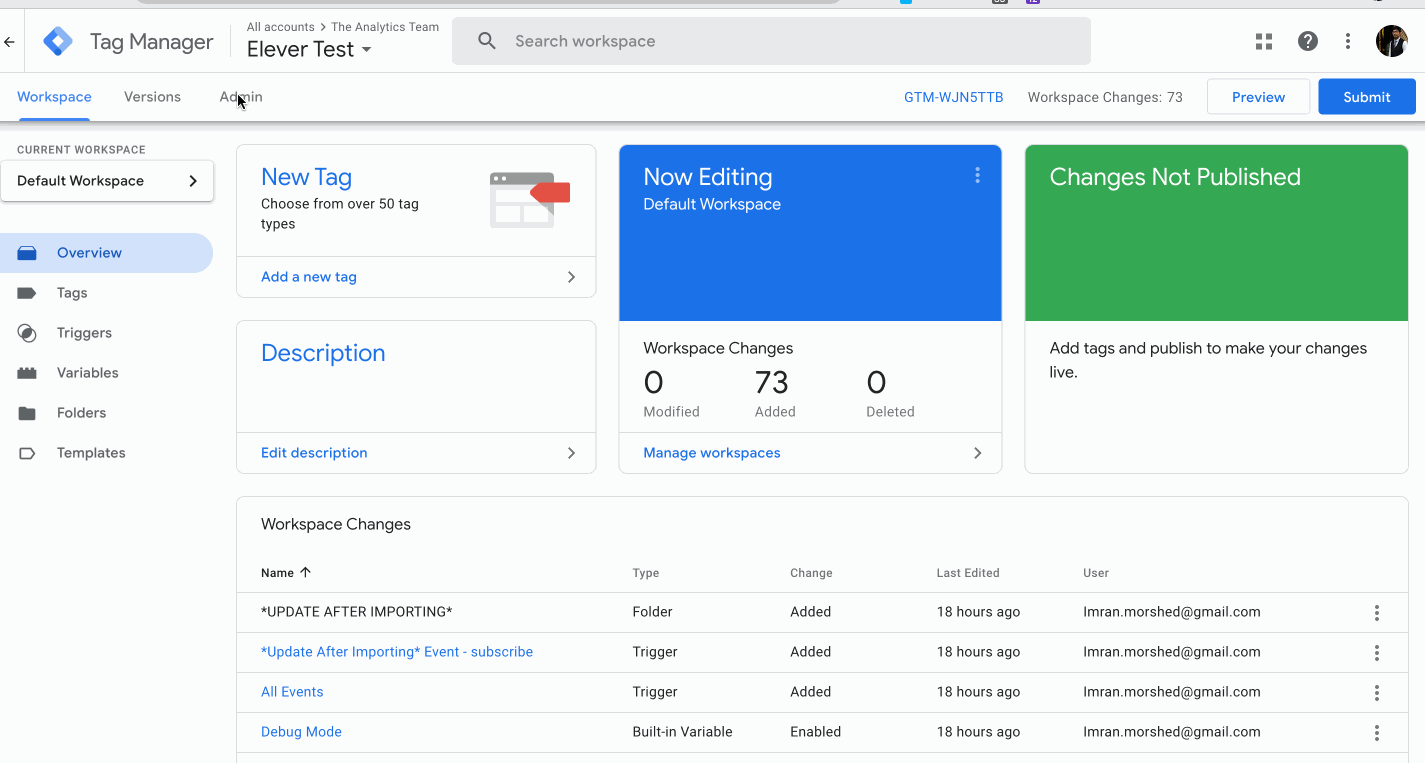
Go to Online Store ----> Themes ----> Snippets ----> Add a new snippet

Name: **elevar-head-listener**

| <script id="elevar-dl-listener-config" type="application/json">{"data\_layer\_listener\_script": "https://shopify-gtm-suite.getelevar.com/shops/d0b0038eba47fc05bb116d997e301563dd42b6d5/3.1.10/events.js", "ss\_url": null, "signing\_key": null, "myshopify\_domain": null}</script>  <script>  (() => {  const configElement = document.getElementById("elevar-dl-listener-config");  if (!configElement) {  console.error("Elevar: DLL Config element not found");  return;  }  const config = JSON.parse(configElement.textContent);  const script = document.createElement("script");  script.type = "text/javascript";  script.src = config.data\_layer\_listener\_script;  script.async = false;  script.defer = true;  script.onerror = function () {  console.error("Elevar: DLL JS script failed to load");  };  script.onload = function () {  if (!window.ElevarGtmSuiteListener) {  console.error("Elevar: `ElevarGtmSuiteListener` is not defined");  return;  }  window.ElevarGtmSuiteListener.handlers.listen({  ssUrl: config.ss\_url,  signingKey: config.signing\_key,  myshopifyDomain: config.myshopify\_domain  });  };  document.head.appendChild(script);  })();  </script> |
| --- |

Now You need to create GTM tags, triggers & variables.

Please [cleck here](https://drive.google.com/file/d/1wku_w0IVZcn7Q4yXCmJLa1Aa4RCbGSfa/view?usp=share_link) to download the full GTM container for GA4.



Please [cleck here](https://drive.google.com/file/d/12i--cvJmvPHcxXR6o7I0xvmfKs9p8TO_/view?usp=share_link) to download the GTM container for Facebook conversion.