**Notes: Exploring Ingested Datasets in Elasticsearch**

**Introduction**

Now that we have ingested our test data into Elasticsearch, let’s take a short moment to talk about it. You already know that we will be working with two different datasets: one containing **HTTP access logs**, and one containing **orders**.

**HTTP Access Logs Dataset**

**Overview**

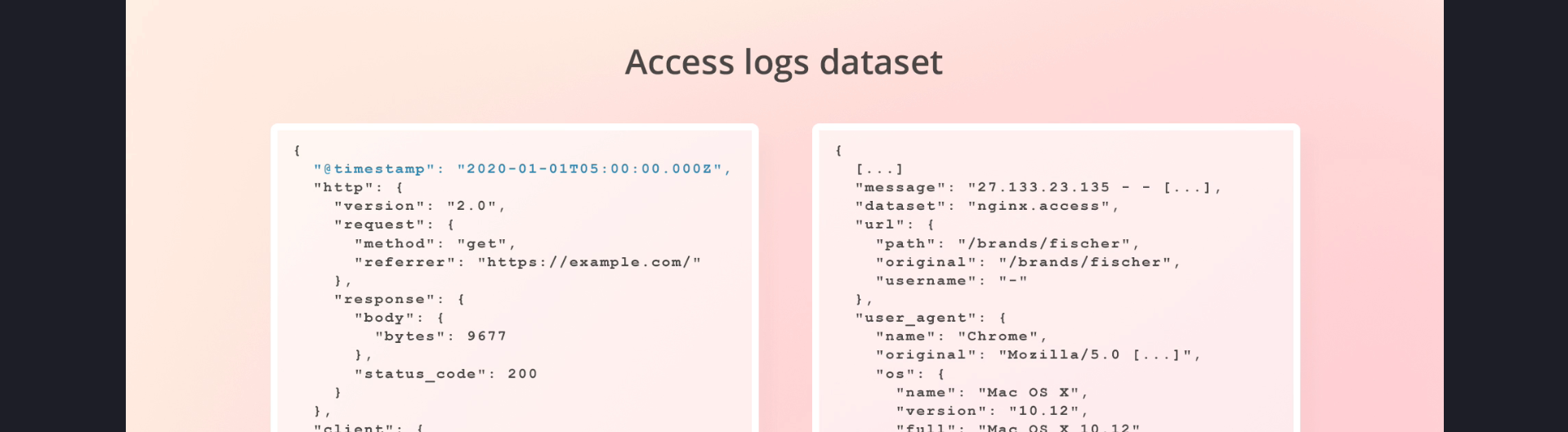
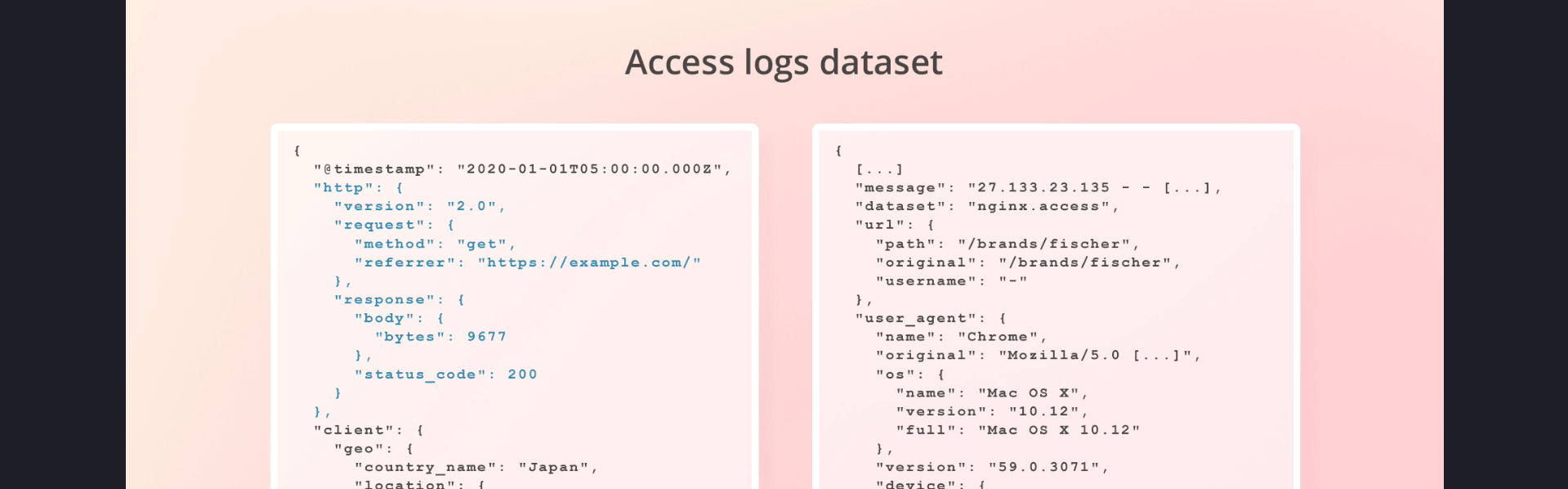
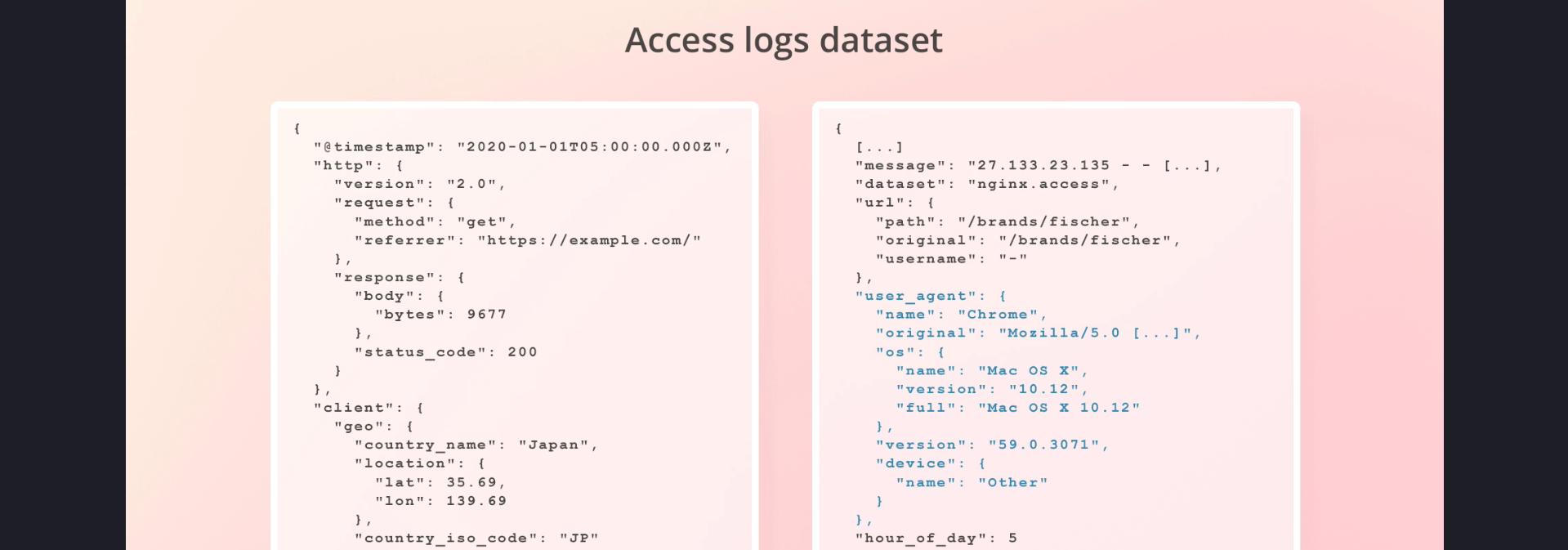
* The HTTP access logs originate from **nginx**, which is commonly used to process HTTP requests.
* The actual HTTP requests are the same whether we use nginx, Apache, or any other web server.
* This is why the structure of our Elasticsearch documents is independent of the web server serving the requests.

**Elastic Common Schema (ECS)**

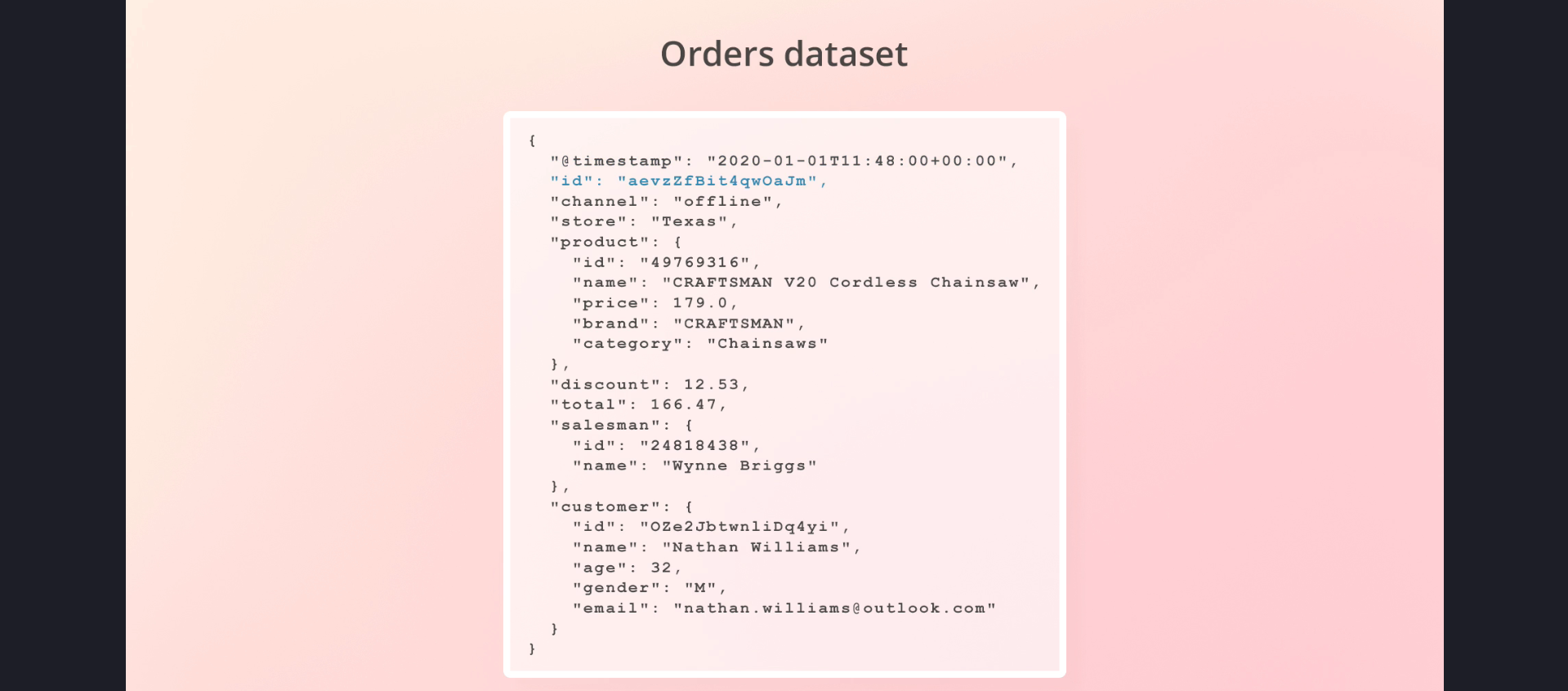
* As mentioned earlier, the **Elastic Common Schema (ECS)** was used when we added the two index templates.
* The index template for the access logs dataset specified field mappings that are compatible with ECS.
* The example document shown on the screen illustrates this structure.

**Document Structure**

* Although you see two JSON objects, they represent the same document; the document was split to fit the screen.
* Let’s quickly run through the fields to understand the data:

1. **@timestamp Field**
   * Contains the time of the event (when the HTTP request was received).
   * 
2. **http Field**
   * 
   * Contains information about the HTTP request, such as:
     + Request method
     + Referrer
     + Status code
3. **client Object**
   * Contains information about the client that sent the HTTP request:
     + **client.ip**: Specifically contains IP addresses when the address resolves to an IP.
     + **client.address**: May contain an IP address, domain, or UNIX socket.
     + **Geographical Information**: Resolved using the **MaxMind GeoLite2 database**.
       - Note: Geolocation is not highly precise with free datasets but is sufficient for this use case.
   * **Why is IP Address Duplicated?**
     + A technicality in ECS specifies different fields for the client’s address and IP.
     + If the address were a domain, the client.domain field would be present alongside client.address.
4. **message Field**
   * Contains the raw event (log line) from which all other fields were extracted.
   * This is the line that was read from within the access log file.
   * Specific to the nginx web server format because web server may use different log formats.
   * Kept for backup purposes to reprocess requests if needed.  
     If we discover any issue with how we processed requests, we are able to use this field to process all the requests again after fixing the issue.
5. **dataset Field**
   * 
   * Specifies the source of the event:
     + Example: "nginx.access" for access logs.
     + **Could be "nginx.error" for error logs.**
6. **url Object**
   * 
   * Contains information about the requested URL:
     + **url.path**: Only the URL path.
     + **url.original**: The unmodified URL, including query parameters.
   * ECS technicality differentiates between these fields.
7. **user\_agent Object**
   * 
   * Contains information about the client’s user agent:
     + **Examples**:
       - Browser (e.g., Chrome, Firefox).
       - HTTP clients like Postman or cURL.
     + Includes details such as:
       - Name
       - Version
       - Device
       - Operating System (OS)
     + The original user agent is also stored.
8. **hour\_of\_day Field**
   * A custom field not defined in ECS.
   * Will be discussed later when talking about heat maps.

**Orders Dataset**



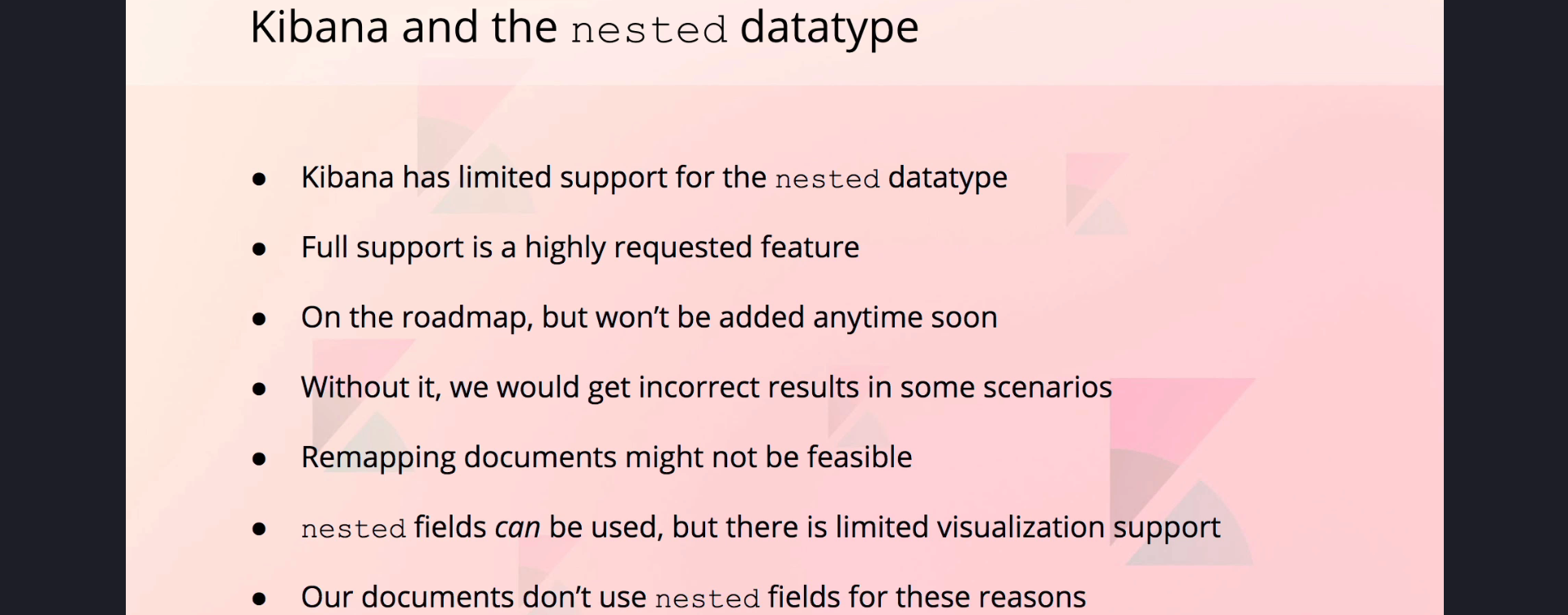
**Overview**

* This dataset contains information about **orders**.
* Similar to the access logs dataset, it uses the @timestamp field from ECS.
* Using ECS for the timestamp field is a good standard but not a strict requirement.

**Fields in the Dataset**

1. **@timestamp Field**
   * Indicates when the order was placed.
2. **id Field**
   * Unique identifier for each order.
3. **channel Field**
   * Specifies whether the order was placed **online** or **offline**.
4. **store Field**
   * Present only for offline orders (e.g., orders placed in physical stores).
5. **product Object**
   * Contains details about the purchased product.
   * **Why Only One Product Per Order?**
     + Technically, multiple products can be defined in an array, as Elasticsearch fields can have multiple values.
     + However, defining products as an array would require using the **nested** datatype to query them independently.
     + **Kibana Limitation**:
       - Kibana lacks robust support for nested fields in visualizations.
       - Running aggregations on nested fields could yield incorrect results.
     + To work around this limitation, the dataset is structured with one product per order.
6. **discount Field**
   * Indicates any discount applied to the order.
   * Not present for all documents.
7. **total Field**
   * Represents the product price after applying any discounts.
8. **salesman Object**
   * Specifies the salesperson responsible for the sale.
   * Present only for offline sales (when channel equals offline).
9. **customer Object**
   * Contains information about the customer who placed the order.
   * Enables analysis of customer demographics.

**Takeaways**



1. **Kibana and Nested Fields**:
   * While nested fields are supported in Elasticsearch, Kibana has limitations in working with them for visualizations.
   * As a workaround, the dataset avoids using nested fields, allowing for simpler visualizations.
2. **Field Mappings**:
   * The field mappings for both datasets were defined in the index templates.
   * These mappings were applied automatically during data ingestion.
3. **Use Cases**:
   * The ingested data can now be used for:
     + Analyzing HTTP request patterns.
     + Visualizing order trends and customer demographics.

**Conclusion**

* We have now explored the structure and fields of both datasets: HTTP access logs and orders.
* The data is ingested into Elasticsearch and ready for use in Kibana.
* In the next steps, we will dive deeper into how to create meaningful visualizations and analyses using this data.