# Detailed Notes on Inspecting an Elasticsearch Cluster

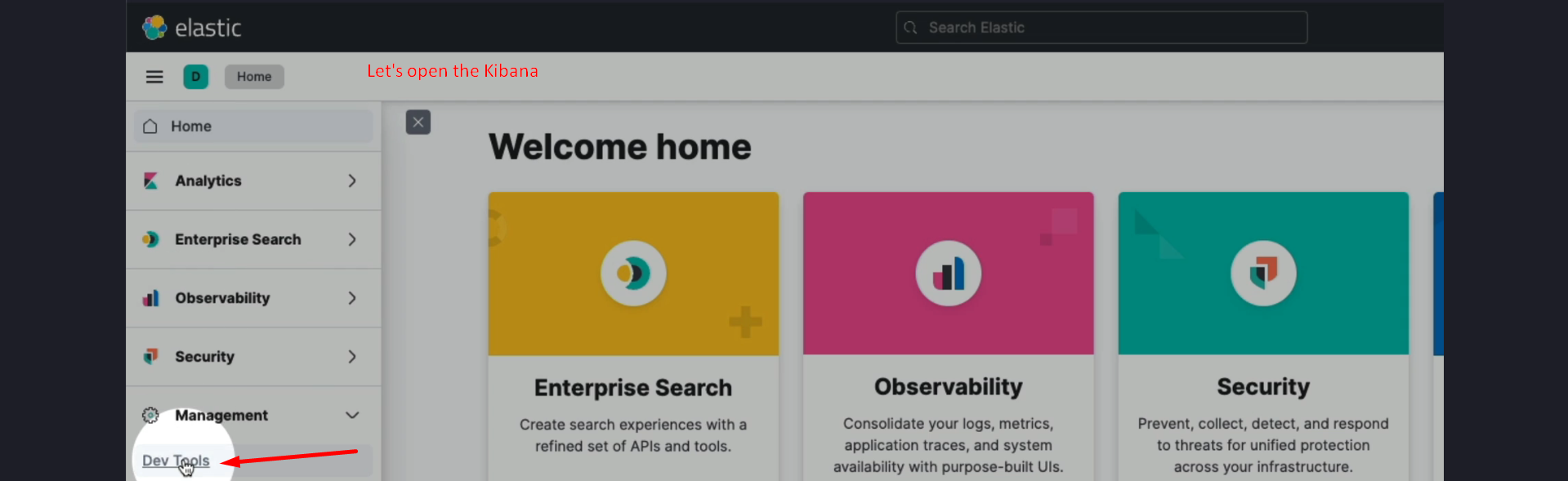
## 

## 1. Introduction to Elasticsearch Cluster Basics

An Elasticsearch cluster is a collection of one or more nodes working together to store and manage data. Understanding the cluster's internal workings involves:  
- Learning how to inspect the cluster using Kibana's Console tool.  
- Interacting with Elasticsearch's REST API for performing actions.

## 2. Using Kibana’s Console Tool

Steps to Access the Console Tool:

1. 1. Open Kibana.  
   2. Expand the menu and click on Dev Tools under the Management section.  
   

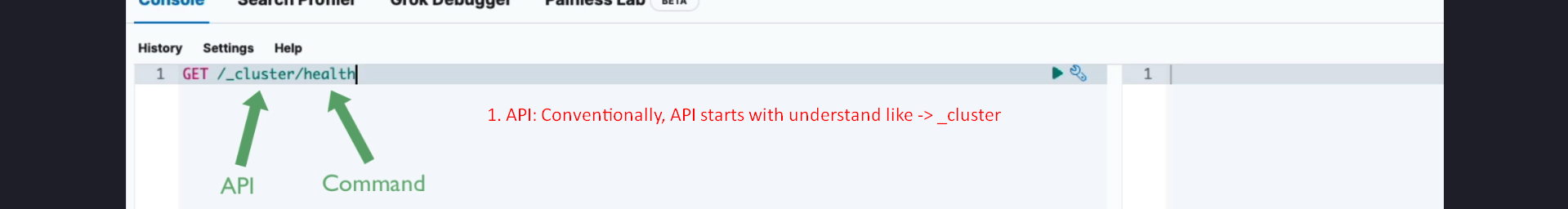
The Console tool allows users to send HTTP requests to Elasticsearch's REST API.

## 3. Retrieving Cluster Health

Query Format:

1. 1. Specify the HTTP verb (e.g., GET).  
   2. Specify the request path, such as /\_cluster/health.  
   - Note: The leading forward slash (/) is optional.

Example Query:

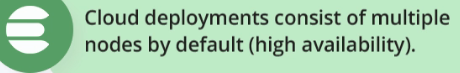
GET /\_cluster/health  


This query retrieves the health status of the cluster.  
  
Result Interpretation:  
- The response is a JSON object.  
- Example fields in the response:  
 - Cluster Name: Default is “elasticsearch”; configurable in elasticsearch.yml.  
 - Status: Indicates cluster health, with "green" meaning the cluster is healthy.  
 NOTE: We will see cases where “status” is not “green”.

## 4. Listing Cluster Nodes

Using the CAT API (**C**ompact and **A**ligned **T**ext):

1. 1. Command: GET /\_cat/nodes?v  
   - The v parameter adds a descriptive header row to the output.

Output includes:  
- Node IP address  
- Node name  
- Performance measures  
  


## 5. Checking Indices in the Cluster

Using the CAT API:

1. 1. Command: GET /\_cat/indices  
   - By default, no indices are shown unless data has been added.

System Indices:  
- Hidden by default and used by Elasticsearch and Kibana for internal storage.  
- Example: Kibana stores dashboards and configuration data within these indices.  
- To view system indices, use the expand\_wildcards query parameter.

Query Example:

GET /\_cat/indices?expand\_wildcards=all

## 6. Key Concepts and Tools

HTTP Requests:

- HTTP requests to Elasticsearch are independent of the client used (Kibana, cURL, etc.).  
- Valid requests access the REST API seamlessly.

Query Parameters:

- Enhance the detail of API responses.  
- Example: Adding the v parameter for headers or expand\_wildcards for hidden indices.

System Indices:

- Identified by a leading period (e.g., .kibana).  
- Comparable to hidden files in Linux/macOS.

## 7. Visual Flow of Operations

Sequence Diagram:

1. 1. User sends a query through Kibana Console.  
   2. Kibana uses Elasticsearch's REST API under the hood.  
   3. Elasticsearch processes the HTTP request and returns a JSON response.

Flow Diagram:

- Input: HTTP verb and path (e.g., GET /\_cluster/health).  
- Processing: Elasticsearch evaluates the request.  
- Output: JSON object with relevant details (e.g., cluster status, node info).

## 8. Summary

Kibana's Console tool provides a user-friendly way to interact with Elasticsearch.  
  
Key APIs include:  
- Cluster Health API: Monitors cluster status.  
- CAT API: Lists human-readable data.  
- Nodes API: Offers detailed node insights.  
  
Understanding these APIs lays the foundation for managing Elasticsearch effectively.