What do you know about Kibana? Explain different elements of Kibana.

🔷 **What is Kibana?**

- **Kibana** is the **visualization and UI tool** for Elasticsearch.
- It lets you **search, explore, visualize, and monitor** data stored in Elasticsearch.
- Instead of writing only APIs, Kibana gives a **web interface** for logs, dashboards, and alerts.

👉 **Analogy:** If Elasticsearch is the **engine** 🚂 storing data, Kibana is the **driver's dashboard** 👨‍✈️ that shows speed, warnings, and trends.

🔷 **Key Elements of Kibana**

1. **Discover** 🔍
   - Search and explore raw data (logs, JSON docs).
   - You can filter logs, sort by timestamp, and drill down into details.
   - Example: Find all "500 Internal Server Error" logs from yesterday.
2. **Visualize** 📊
   - Create charts (bar, pie, line, heatmaps) from data.
   - Example: Show number of failed logins per hour.
3. **Dashboard** 🗂️
   - Combine multiple visualizations into one screen.
   - Example: A "Web App Monitoring" dashboard showing CPU usage, error rates, and user traffic in one view.
4. **Filter & Search**
   - Use **KQL (Kibana Query Language)** or filters to refine data.
   - Example: status:500 AND path:"/login" → shows only failed login requests.
5. **Alerts & Watchers** ⏰
   - **Alerts** (in recent Kibana versions) → Set conditions to notify when metrics/logs cross thresholds.
   - Example: Alert if error rate > 100/min.

- **Watcher** (part of X-Pack in older versions) → More advanced alerting/automation.

🔷 **How to Create a Visualization or Dashboard in Kibana**

1. Go to **Visualize** → select chart type (e.g., line chart).
2. Choose an **index pattern** (like nginx-logs*).
3. Pick fields → (e.g., @timestamp for X-axis, status_code count for Y-axis).
4. Save the visualization.
5. Add it to a **Dashboard** → Combine multiple visualizations.

👉 Example Dashboard:

- Line chart → Requests per second.
- Pie chart → Status code distribution (200/400/500).
- Table → Top 10 IP addresses with most requests.

🔷 **Example Scenario (How You'd Use It in DevOps)**

Imagine your app is slow:

1. In **Discover**, filter for status:500.
2. See spikes in errors around 2:15 PM.
3. Go to **Dashboard** → error rate chart shows a big spike.
4. Drill down into **Visualize** → find most errors from /checkout API.
5. Set up an **Alert** → "Send Slack notification if checkout errors > 50/min."

🔷 **Short Interview Answer**

- Kibana = **UI & visualization tool** for Elasticsearch.
- Discover and explore **raw logs**.
- Create **charts, graphs, dashboards**.
- Use **Kibana Query Language (KQL)** to filter/search.
- Set up **alerts/watchers** for monitoring.
- Example: dashboard for errors, performance, traffic + alerts on error spikes.

What is the primary purpose of **Kibana** in the ELK stack?
A) Store logs and metrics
B) Collect logs from different servers
C) Visualize and explore Elasticsearch data
D) Index raw JSON documents
Answer: C) Visualize and explore Elasticsearch data

Which **Kibana feature** allows you to search and explore raw logs stored in Elasticsearch?
A) Dashboard
B) Visualize

C) Discover
D) Alerts
Answer: C) Discover

If you want to create a **line chart of requests per second**, which Kibana element would you use?
A) Discover
B) Visualize
C) Dashboard
D) Filter
Answer: B) Visualize

What is the purpose of a **Dashboard** in Kibana?
A) Store documents as JSON
B) Combine multiple visualizations into one view
C) Ingest logs from syslog and Beats
D) Generate cluster health reports
Answer: B) Combine multiple visualizations into one view

Which query language does Kibana provide for advanced filtering?
A) SQL
B) KQL (Kibana Query Language)
C) Grok
D) JSONPath
Answer: B) KQL (Kibana Query Language)

What can you do with Kibana Alerts?
A) Build new indices
B) Trigger notifications when conditions are met
C) Ingest logs from Kafka
D) Split indices into shards
Answer: B) Trigger notifications when conditions are met
