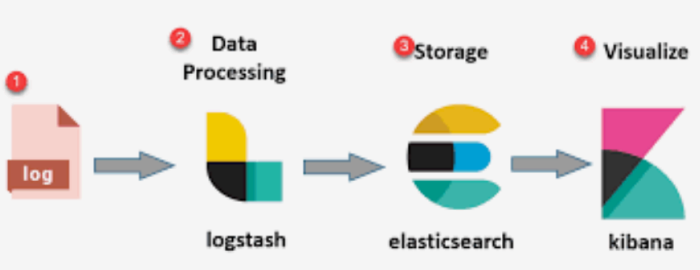
What is ELK Stack, and why is it used in DevOps?

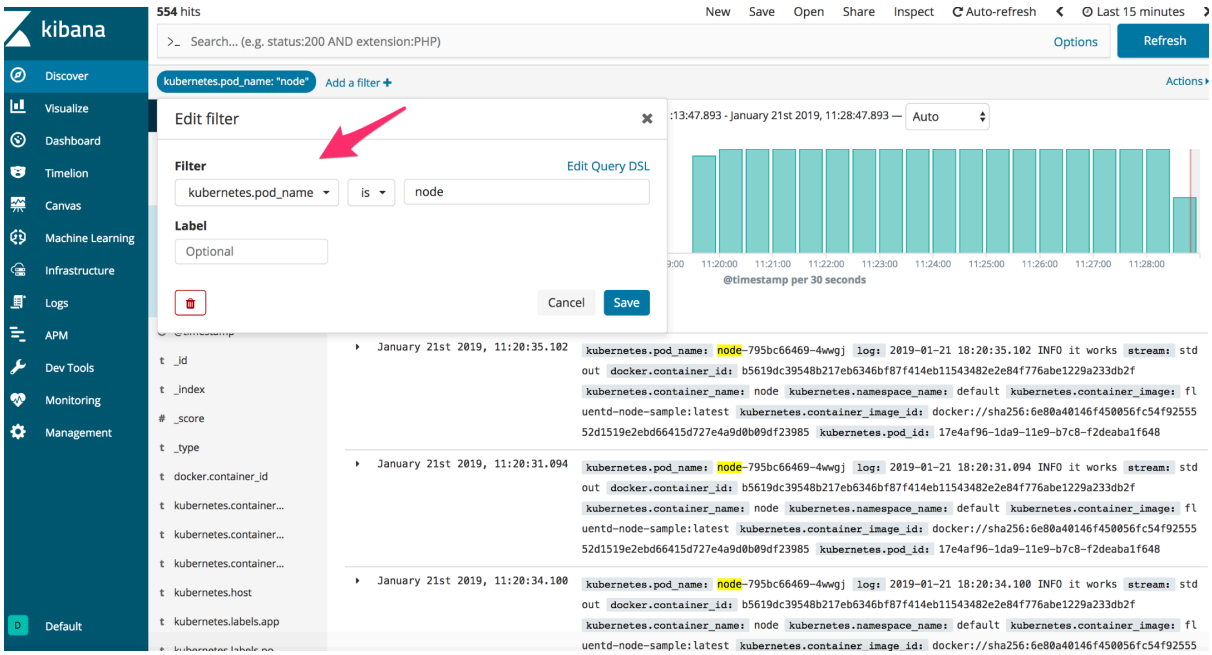
🔷 What is ELK Stack?

ELK = **Elasticsearch + Logstash + Kibana**

- **Elasticsearch** → A search & analytics engine (stores and searches logs really fast).
- **Logstash** → A pipeline tool (collects logs from different sources → cleans/transforms → sends to Elasticsearch).
- **Kibana** → A dashboard tool (visualizes the data from Elasticsearch in graphs, charts, dashboards).

👉 Together, they form the **ELK Stack** – a popular log management and monitoring solution.





🔷 Why is ELK used in DevOps?

In DevOps, we deal with **lots of servers, containers, applications**. Each produces tons of logs 📜 (errors, performance data, events).
Without ELK:

- Logs are scattered across many servers.
- Debugging takes forever ("Where's that error coming from?").

With ELK:

1. **Centralized logging** – All logs in one place.
2. **Searchable** – Elasticsearch lets you find "error 500" in seconds across all servers.
3. **Visualization** – Kibana dashboards show error trends, request rates, performance.
4. **Troubleshooting faster** – Instead of SSH-ing into 10 servers, you just check Kibana.

🔷 **Use Cases in DevOps**

- **Monitoring application errors** (find root cause quickly).
- **Analyzing performance** (response times, throughput).
- **Security auditing** (failed login attempts, suspicious activity).
- **Compliance** (track system activity for regulations).

👉 **One-liner answer (if interviewer wants short):**
"ELK Stack is Elasticsearch, Logstash, and Kibana – used in DevOps for centralized logging, searching, and visualizing logs. It helps engineers troubleshoot faster, monitor performance, and ensure reliability across systems.

System Setup:

Updated the Amazon Linux 2 system and installed Docker
Installed Docker Compose to manage multiple containers
Set up user permissions for Docker access
ELK Stack Deployment:

INSTALL DOCKER ON AMAZON LINUX 2

```
sudo yum update -y
sudo amazon-linux-extras enable docker
sudo yum install -y docker
sudo systemctl start docker
sudo systemctl enable docker
sudo usermod -aG docker ec2-user
```

DOCKER VERIFICATION

```
docker --version
docker ps
```

INSTALL DOCKER COMPOSE

sudo curl -L "https://github.com/docker/compose/releases/download/v2.27.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
docker-compose --version

mkdir elk && cd elk

Created a Docker Compose configuration to run three main services:
Elasticsearch: A search engine that stores and indexes log data
Kibana: A web dashboard for visualizing and searching through logs
Logstash: A data processing pipeline that receives and transforms logs
Configured these services to work together in a network
Started all services as background containers
---

docker-compose.yml

CREATE THE YML FILE

```
version: "3.7"
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:8.15.0
    container_name: elasticsearch
    environment:
      - discovery.type=single-node
      - xpack.security.enabled=false   # 🔷 Disabled security for demo
      - ES_JAVA_OPTS=-Xms512m -Xmx512m
    ports:
      - "9200:9200"
    networks:
      - elk

  kibana:
    image: docker.elastic.co/kibana/kibana:8.15.0
    container_name: kibana
    ports:
      - "5601:5601"
    environment:
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
    depends_on:
```

```
    - elasticsearch
   networks:
    - elk

  logstash:
   image: docker.elastic.co/logstash/logstash:8.15.0
   container_name: logstash
   ports:
    - "5044:5044"
   volumes:
    - ./logstash.conf:/usr/share/logstash/pipeline/logstash.conf
   depends_on:
    - elasticsearch
   networks:
    - elk

networks:
  elk:
   driver: bridge
```

CREATE THE logstash.conf FILE

```
input {
  beats {
   port => 5044
  }
}

output {
  elasticsearch {
   hosts => ["http://elasticsearch:9200"]
   index => "python-logs"
  }
}
```

START THE ELK STACK

```
docker-compose up -d
docker ps
```

NOW USE ----->>> public ip and the two ports which are 9200 (for ElastiSearch API) and 5601(for Kibana Dashboard)

WHEN docker ps is RUN ---> CONTAINER ID IMAGE ETC WILL SHOW

AFTER ACCESSING KIBANA UI ---> CREATE AN INDEX PATTERN

sudo docker-compose up -d logstash (MAKE SURE ELASTISEARCH AND CONTAINER LOGSTASH IS RUNNING)

CONFIGURE KIBANA

http://elasticsearch:9200

Created a simple Python application that generates log messages
Set up the app to write timestamped log entries to a file on the system
Ran the application to create sample log data
Log Collection:

CREATE A PYTHON FILE TO LOG

mkdir -p ~/python-app
cd ~/python-app

nano app.py

```
import logging
import time

logging.basicConfig(
   filename="/home/ec2-user/python-app.log",
   level=logging.INFO,
   format="%(asctime)s - %(levelname)s - %(message)s"
)

for i in range(20):
   logging.info(f"Hello from Python EC2 app iteration {i}")
   time.sleep(2)
```

python3 ~/python-app/app.py

INSTALL FILEBEAT ON SYSTEM

curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.15.0-x86_64.rpm
sudo rpm -vi filebeat-8.15.0-x86_64.rpm

sudo nano /etc/filebeat/filebeat.yml

```
# ============================== Filebeat inputs ===============================

filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - /home/ec2-user/python-app.log
    # Optional: multiline if your logs span multiple lines
    # multiline.pattern: '^\d{4}-\d{2}-\d{2}'
    # multiline.negate: true
    # multiline.match: after

# ============================== Filebeat modules ==============================
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false

# ======================= Output to Logstash =======================
output.logstash:
  # The Logstash host and port
  hosts: ["localhost:5044"]

# ======================= Optional: Logging =======================
logging.level: info
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat.log
  keepfiles: 7
  permissions: 0644
```

========== Optional: Internal metrics ==========
monitoring.enabled: false

Installed Filebeat, a lightweight log shipping agent
Configured Filebeat to monitor the Python application's log file
Set up Filebeat to send collected logs to Logstash for processing
Final Result: The system now automatically collects logs from the Python application, processes them through the ELK stack, and makes them available for searching and visualization in the Kibana web interface. Users can access Kibana through a web browser to view, search, and analyze the logs in real-time.

sudo systemctl enable filebeat
sudo systemctl start filebeat

NOW CAN SEE THE LOGS AS WELL

Which ELK component is responsible for collecting and transforming logs before sending them to storage?
A) Elasticsearch
B) Kibana
C) Logstash
D) Fluentd
✅ **Answer:** C) Logstash

In DevOps, what is the main benefit of using ELK for logs?
A) It automatically fixes errors in applications
B) It centralizes logs, makes them searchable, and visualizes trends
C) It replaces the need for monitoring tools
D) It removes the need for debugging
✅ **Answer:** B) It centralizes logs, makes them searchable, and visualizes trends

---