

**document this**

Here's the complete documented process for setting up ELK stack (Elasticsearch, Logstash, Kibana) with Docker Compose, integrating Spring Boot logs collection via Filebeat, detailing errors encountered and how they were fixed:

**Explain the tools**

* **Docker** – We run our services (Spring Boot, Elasticsearch, Logstash, Filebeat) in containers.
* **Filebeat** – An agent installed inside the Docker host that reads logs from container stdout/stderr and sends them to Logstash.
* **Logstash** – Processes and structures logs; applies patterns (Grok) and sends them to Elasticsearch.
* **Elasticsearch** – Stores all logs in indexes, allows searching and aggregation.
* **Kibana** – Visualizes logs, allows filtering errors, warnings, or info messages.

**3️⃣ Explain the data flow**

1. Spring Boot writes logs to **console** (stdout/stderr) or log file in container.
2. **Filebeat** watches /var/lib/docker/containers/\*/\*.log and harvests all new log lines.
3. Filebeat sends logs to **Logstash** over port 5044.
4. Logstash tries to parse logs using **Grok patterns**.
   * If Grok fails, the log is stored raw in event.original.
5. Parsed logs are sent to **Elasticsearch** and stored in daily indices, e.g., project-logs-2025.08.27.
6. **Kibana** can be used to search or visualize logs, filter by log level (ERROR) or keywords (Exception).

**4️⃣ Show how runtime errors appear**

* In Kibana, search event.original for ERROR or Exception.
* Example: Spring Boot stack trace will appear in event.original until a proper Grok pattern is applied.

https://chatgpt.com/share/68aedd05-c8fc-800a-bb86-f4b526e90c9d

**ELK Stack with Docker and Spring Boot Logs: Full Setup Documentation**

**1. Objective**

To set up a centralized logging and monitoring system for a Spring Boot application running in Docker using the ELK stack (Elasticsearch, Logstash, Kibana) with Filebeat for log shipping and to visualize logs via Kibana dashboards.

**2. Environment Details**

* **OS**: Ubuntu (Azure VM)
* **Docker**: Version 24.x
* **Docker Compose**: Version 2.x
* **ELK Stack Version**: 8.10.0
* **Services**:
  + Spring Boot backend app
  + MySQL database
  + Telegram bot
  + Watchtower (auto-update Docker images)
  + ELK components (Elasticsearch, Logstash, Kibana, Filebeat)

**3. Docker Compose Setup**

**docker-compose.yml snippet**

version: "3.9"  
  
services:  
 elasticsearch:  
 image: docker.elastic.co/elasticsearch/elasticsearch:8.10.0  
 container\_name: elasticsearch  
 environment:  
 - discovery.type=single-node  
 - ES\_JAVA\_OPTS=-Xms512m -Xmx512m  
 - xpack.security.enabled=false  
 ports:  
 - "9200:9200"  
 - "9300:9300"  
 networks:  
 - elk-network  
 volumes:  
 - elasticsearch\_data:/usr/share/elasticsearch/data  
  
 kibana:  
 image: docker.elastic.co/kibana/kibana:8.10.0  
 container\_name: kibana  
 ports:  
 - "5601:5601"  
 depends\_on:  
 - elasticsearch  
 networks:  
 - elk-network  
  
 logstash:  
 image: docker.elastic.co/logstash/logstash:8.10.0  
 container\_name: logstash  
 ports:  
 - "5044:5044"  
 volumes:  
 - /home/azureuser/logstash.conf:/usr/share/logstash/pipeline/logstash.conf:ro  
 - springboot\_logs:/home/azureuser/springboot\_logs:ro  
 depends\_on:  
 - elasticsearch  
 networks:  
 - elk-network  
  
 filebeat:  
 image: docker.elastic.co/beats/filebeat:8.10.0  
 container\_name: filebeat  
 user: root  
 volumes:  
 - /var/lib/docker/containers:/var/lib/docker/containers:ro  
 - /var/run/docker.sock:/var/run/docker.sock:ro  
 - ./filebeat.yml:/usr/share/filebeat/filebeat.yml:ro  
 depends\_on:  
 - logstash  
 networks:  
 - elk-network  
  
 final-year-backend:  
 image: pasindu55/final-year-project:latest  
 container\_name: final-year-backend  
 environment:  
 SPRING\_DATASOURCE\_URL: jdbc:mysql://final-year-db:3306/${MYSQL\_DATABASE}?createDatabaseIfNotExist=true  
 SPRING\_DATASOURCE\_USERNAME: ${MYSQL\_USER}  
 SPRING\_DATASOURCE\_PASSWORD: ${MYSQL\_PASSWORD}  
 ports:  
 - "8080:8080"  
 volumes:  
 - springboot\_logs:/home/azureuser/springboot\_logs  
 networks:  
 - backend-network  
  
volumes:  
 elasticsearch\_data:  
 springboot\_logs:  
  
networks:  
 elk-network:  
 backend-network:

**4. Filebeat Configuration**

**filebeat.yml**

filebeat.inputs:  
- type: log  
 enabled: true  
 paths:  
 - /var/lib/docker/containers/\*/\*.log  
 json.keys\_under\_root: true  
 json.add\_error\_key: true  
  
output.logstash:  
 hosts: ["logstash:5044"]

* Collects Docker container logs in JSON format.
* Sends logs to Logstash on port 5044.
* Filebeat runs as root user, so the filebeat.yml permissions must be correct.

**5. Logstash Configuration**

**logstash.conf**

input {  
 beats {  
 port => 5044  
 }  
}  
  
filter {  
 # Initially: grok filter failed due to mismatch with Spring Boot JSON logs  
 # Changed to parse JSON logs correctly:  
 json {  
 source => "message"  
 target => "springboot"  
 remove\_field => ["message"]  
 }  
 # Optional: rename or extract fields for easier use in Kibana dashboards  
}  
  
output {  
 elasticsearch {  
 hosts => ["http://elasticsearch:9200"]  
 index => "project-logs-%{+YYYY.MM.dd}"  
 }  
 stdout { codec => rubydebug } # for debugging  
}

* Listens for Beats input on port 5044.
* Parses JSON logs sent by Filebeat from Docker container logs.
* Sends data to Elasticsearch with daily indices.

**6. Key Steps, Errors, and Fixes**

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Issue Description | Cause | Fix / Resolution |
| Initial Docker Compose startup | Warning: version attribute obsolete in docker-compose.yml | Old docker-compose version attribute | Removed or updated version in docker-compose.yml |
| Filebeat container fails to start | Mount error: trying to mount directory onto file path (filebeat.yml) | filebeat.yml was a directory, not a file | Removed directory, created actual file filebeat.yml with correct config |
| Filebeat config file ownership error | Filebeat refuses to load config: owner must be root (uid=0) | filebeat.yml lacked root ownership | Changed filebeat.yml owner: sudo chown root:root filebeat.yml and set permissions chmod 644 |
| Filebeat cannot connect to Logstash | connection refused error to logstash:5044 | Logstash not listening on port 5044 or not running | Exposed port 5044 in docker-compose, verified logstash.conf has beats input, restarted containers |
| Logstash pipeline shuts down immediately | Pipeline terminated and Logstash shut down | Wrong or malformed logstash.conf, missing input or output, or permission issues | Fixed logstash.conf with minimal input-output pipeline and ensured config file permissions |
| Logstash \_grokparsefailure | Raw logs not parsed, grok filter mismatched Spring Boot log format | Grok filter pattern did not match JSON logs from Spring Boot | Replaced grok with json { source => "message" } filter to parse JSON logs properly |
| Environment variables not set warnings | Several warnings about MYSQL\_USER, MYSQL\_PASSWORD etc. not set | .env file missing or variables not defined | Created .env file or set environment variables for MySQL database credentials |
| Verified logs in Elasticsearch index | No visible structured fields, raw logs stored | Parsing not correct so fields not extracted | Adjusted logstash.conf JSON filter, confirmed logs indexed in project-logs-YYYY.MM.DD |
| Kibana setup | Created index pattern project-logs-\*, viewed logs in Discover tab | - | Ready to create dashboards and alerts |

**7. How the Data Flows**

* Spring Boot app writes logs to Docker container stdout (JSON format).
* Filebeat reads container logs from /var/lib/docker/containers/\*/\*.log.
* Filebeat forwards logs to Logstash via Beats protocol on port 5044.
* Logstash parses logs using JSON filter and sends structured data to Elasticsearch.
* Elasticsearch indexes logs in daily indices (e.g., project-logs-2025.08.27).
* Kibana provides UI to query, visualize, and create dashboards from the data.

**8. Tips and Recommendations**

* Always ensure your config files (filebeat.yml, logstash.conf) have correct ownership and permissions.
* Use minimal Logstash pipeline first to validate connectivity, then add filters.
* Use JSON format for Spring Boot logs for easier parsing. Configure Spring Boot logging accordingly.
* Define environment variables in a .env file used by Docker Compose to avoid warnings.
* Build Kibana dashboards with relevant visualizations: error rates, log levels, exceptions, request latencies.
* Consider enabling machine learning anomaly detection on log patterns in Kibana (Elastic ML).

**9. Next Steps**

* Create custom Logstash grok filters if Spring Boot logs formatting deviates from JSON.
* Set up retention policies for Elasticsearch indices to manage storage.
* Automate alerts in Kibana for error spikes or anomalies.
* Optionally, add enrichment filters (e.g., GeoIP, user agent parsing).
* Monitor resource usage of ELK stack containers and optimize JVM memory if needed.

This documentation summarizes the whole ELK stack with Docker Compose integration for Spring Boot logs including setup details, errors encountered like file mounting issues, connection refused errors, grok failures, their root causes, and fixes applied.

If a ready-to-use full docker-compose.yml, filebeat.yml, and logstash.conf package with structured Spring Boot log parsing and Kibana dashboard templates is needed, it can be provided for immediate deployment.

Would such a ready-to-run package be helpful?

⁂

1. paste.txt