ELK DEPLOYMENT

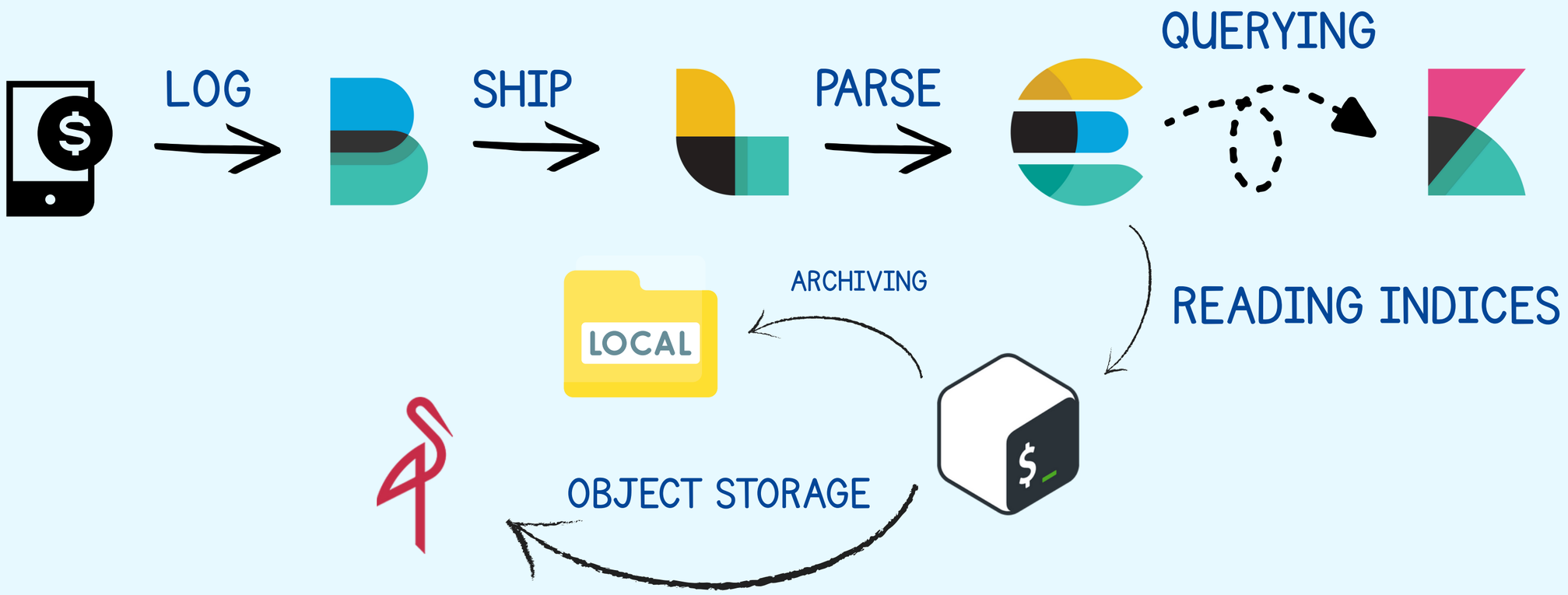
FEATURING

Presented by Amirhossein Ebrahimzade
(who knows how to quit VIM)

WHY ELK?

- In modern infrastructures, logs are valuable but massive
- Problem: Long-term log retention explodes storage costs
- Goal: Build a reliable system to collect, visualize, and archive logs efficiently

WHAT WE HAVE HERE? (HLD)



EDEBEATRCONFIGURATION:

- Common Structure, Different in Details:
  - Common points: id, processtag, timestamp, transaction_id, log_type, details, ...
  - Differences: log_type, process_tag, details,

event, log.rotation.Event, {}

statisticsPartial, Performance.Metric.StatisticsPartial, {some Details}

statisticsPartial, Performance.GC.StatisticsPartial, {some Details}

Metrics (CPU, Memory)

GC

DOCKER COMPOSE:

- Elasticsearch
  - 8.15.0, NoAuth, PV: es_data, Port 9200
- Logstash
  - 8.15.0, Filebeat input, Custom Pipeline, Logstash.conf
- Kibana
  - 8.15.0, GUI, Port Exposed on 5601
- Filebeat
  - 8.15.0, /var/log/*.json Mounted, filebeat.yml
- Elasticdump
  - Custom Image with Local Dockerfile, Dumps indices and Uploads to MinIO, Uses export-and-uplod.sh
- MinIO
  - SelfHosted, Port Exposed on 9000, 9001

SHARED: elk-net(bridge), Health Checks

FILEBEAT CONFIGURATION:

```yaml
filebeat.inputs:
- type: filestream
  id: my-json-logs
  enabled: true
  paths:
    - /var/log/elk/*.json

  parsers:
    - multiline:
        pattern: '^\{'
        negate: true
        match: 'after'

output.logstash:
  hosts: ["logstash:5044"]
```

Instead of “log”, Cause It’s improved.

Looks for initial {, negates following lines, then completes with }

Logstash endpoint for shipping logs

LOGSTASH CONFIGURATION:

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

Input from the Filebeat, on endpoint port 5044

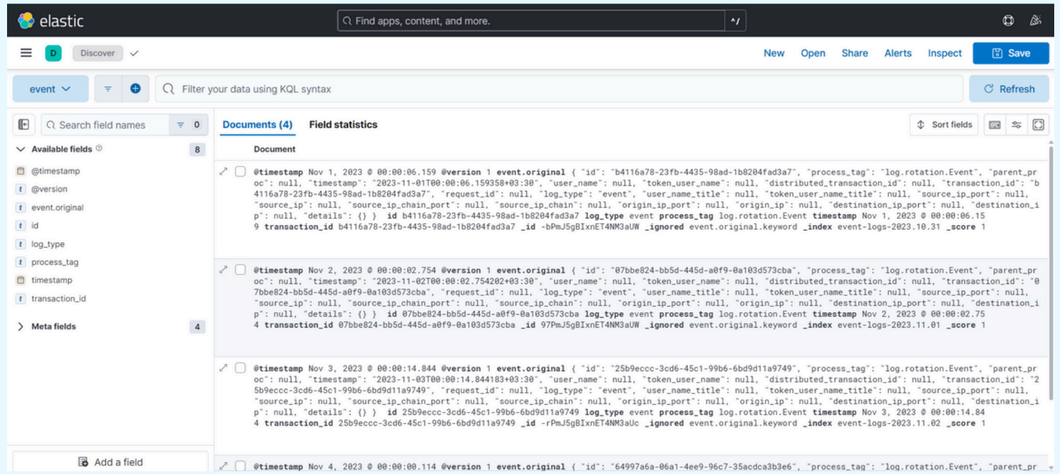
```
filter {
  json {
    source => "message"
  }

  date {
    match => [ "timestamp", "ISO8601" ]
    target => "@timestamp"
  }

  mutate {
    remove_field => [ "agent", "ecs", "host", "log", "input", "tags", "message" ]
  }
}
```

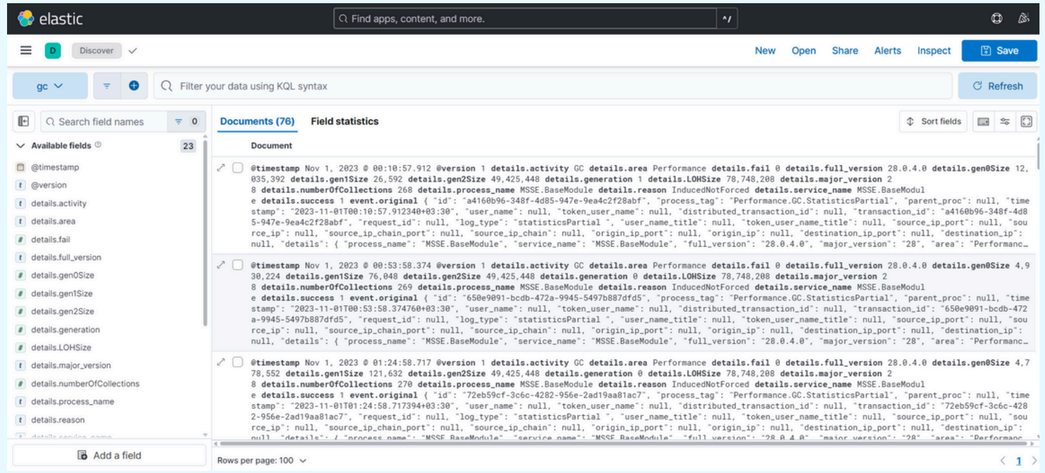
Filtering, Timestamp issues, mutation

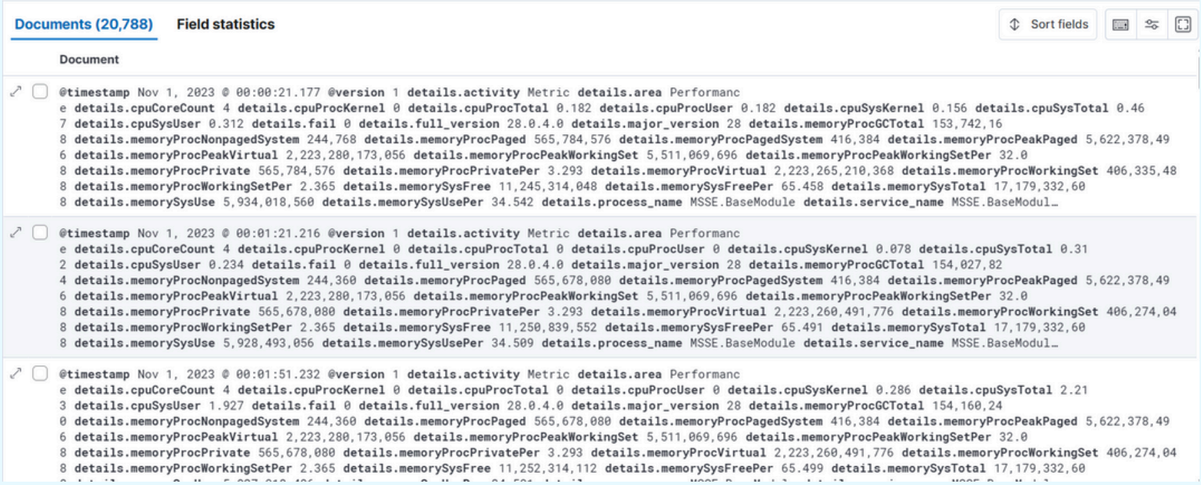
```
output {
  if [process_tag] == "log.rotation.Event" {
    elasticsearch {
      hosts => ["http://elasticsearch:9200"]
      index => "event-logs-%{+YYYY.MM.dd}"
    }
  } else if [process_tag] == "Performance.GC.StatisticsPartial" {
    elasticsearch {
      hosts => ["http://elasticsearch:9200"]
      index => "gc-logs-%{+YYYY.MM.dd}"
    }
  } else if [process_tag] == "Performance.Metric.StatisticsPartial" {
    elasticsearch {
      hosts => ["http://elasticsearch:9200"]
      index => "metric-logs-%{+YYYY.MM.dd}"
    }
  } else {
    elasticsearch {
      hosts => ["http://elasticsearch:9200"]
      index => "unhandled-logs-%{+YYYY.MM.dd}"
    }
  }
}
```

Indices based on “process_tag”, handles unknown logs via unknown “process_tag” in “unhandled-logs-*”

WHAT WE SEE ON KIBANA:







ELASTIC DUMP AND MOREOVER:

Docker file:

```
FROM node:18

RUN npm install -g elasticdump \
 && curl -O https://dl.min.io/client/mc/release/linux-amd64/mc \
 && chmod +x mc && mv mc /usr/local/bin/mc

WORKDIR /dumps

COPY export-and-upload.sh /elasticdump/export-and-upload.sh
RUN chmod +x /elasticdump/export-and-upload.sh

CMD ["/bin/bash", "/elasticdump/export-and-upload.sh"]
```

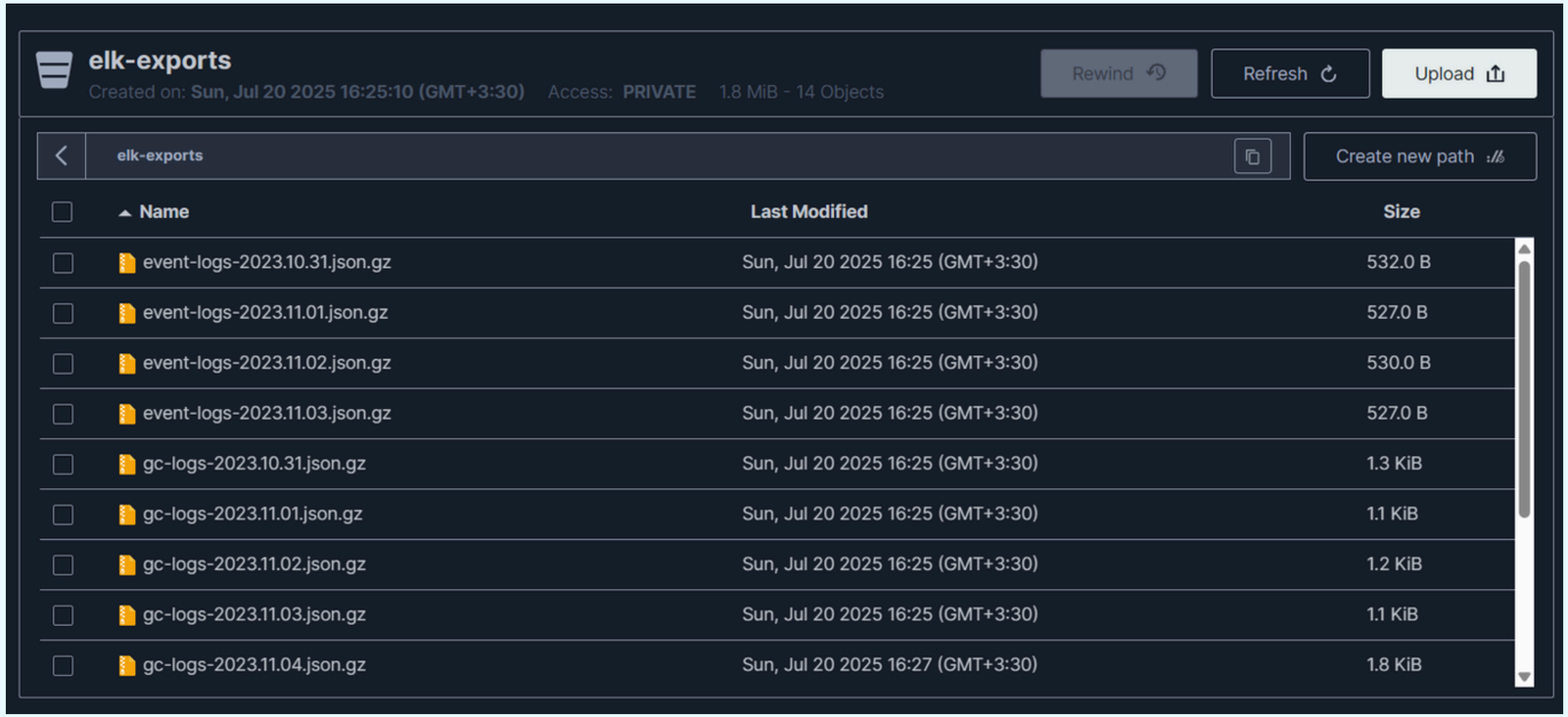
export-and-uplod.sh

```
#!/bin/bash

ES_HOST="http://elasticsearch:9200"
MINIO_BUCKET="elk-exports"
EXPORT_DIR="/dumps"
PATTERNS=("event-logs-" "gc-logs-" "metric-logs-")

export_to_minio() {
  for PREFIX in "${PATTERNS[@]}"; do
    INDICES=$(curl -s "$ES_HOST/_cat/indices?h=index" | grep "^${PREFIX}")

    for INDEX in $INDICES; do
      FILE="$EXPORT_DIR/${INDEX}.json.gz"
      if [[ -f "$FILE" ]]; then
        echo "[SKIP] $INDEX already exported"
        continue
      fi

      echo "[EXPORT] Dumping $INDEX"
      elasticdump --input="${ES_HOST}/${INDEX}" --output="${EXPORT_DIR}/${INDEX}.json" --type=data
      gzip -f "${EXPORT_DIR}/${INDEX}.json"

      echo "[UPLOAD] Uploading $INDEX to MinIO"
      mc alias set minio http://minio:9000 minioadmin minio12345678@
      mc mb -q --ignore-existing minio/$MINIO_BUCKET
      mc cp "$FILE" minio/$MINIO_BUCKET/
    done
  done
}

# Loop forever
while true; do
  export_to_minio
  echo "[WAIT] Sleeping for a minute..."
  sleep 60
done
```

MINIO:

Why Minio?

- S3 Compatibiliy
- Local Backup Storage
- Fast and Easy Deployment
- Reliable & Scalable
- Powerful CLI tool (MC)



FUTURE IMPROVEMENTS (PRODUCTION READY):

- Security Enhancements, env variables or secret managers
- Data Handling & Storage (ILM)
- Alerting Modules
- Move to K8S, Using Helm charts
- Automating Backups using CronJobs

THANKS FOR WATCHING! :D