



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт компьютерных наук
Кафедра автоматизированных систем управления

Лабораторная работа №5
по дисциплине «Теория обработки больших массивов данных»

Студент М-РИТ-25

Станиславчук С. М.

(подпись, дата)

Руководитель

Доцент, к.т.н.

Тюрин А. С.

(подпись, дата)

Липецк 2025

Содержание

1. Задание кафедры.....	2
2. Ход работы.....	3
3. Вывод.....	6
Приложение 1. docker-compose.yml.....	7
Приложение 2. Скрипт bash для генерации логов GET.....	8

1. Задание кафедры

Развернуть elastic и filebeat, проверить работоспособность конфигурации, проверить доступность веб-интерфейса elastic (kibana). Используя скрипт, создать логи, после чего прочитать их, используя filebeat.

Цель: изучить технологию создания и использования очередей сообщения, чтения и записи сообщений в очереди.

2. Ход работы

** В ходе выполнения работы использовалось ядро GNU/Linux v6.15.4, Docker v28.3.0, elastic v8.5.0, kibana v8.5.0, filebeat v8.11.0, Python v3.12.9 **

docker-compose.yml:

```
services:
  elasticsearch:
    image: elasticsearch:8.5.0
    container_name: elasticsearch
    environment:
      - discovery.type=single-node
      - ES_JAVA_OPTS=-Xms512m -Xmx512m
      - xpack.security.enabled=false
      - cluster.routing.allocation.disk.threshold_enabled=false
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - es_data:/usr/share/elasticsearch/data
    ports:
      - "9200:9200"
    networks: [elastic]

  kibana:
    image: kibana:8.5.0
    container_name: kibana
    environment:
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
      - ELASTICSEARCH_REQUESTTIMEOUT=120000
    ports:
      - "5601:5601"
    networks: [elastic]
    depends_on: [elasticsearch]

  filebeat:
    image: elastic/filebeat:8.11.0
    container_name: filebeat
    user: "0"                      # чтобы точно было право читать файлы
    environment:
      - STRICT_PERMS=false
    volumes:
```

```

- ./filebeat.yml:/usr/share/filebeat/filebeat.yml:ro
- ./logs:/logs:ro          # <-- монтируем папку с 1.log
# Если не читаем логи контейнеров – эти два монтирования можно убрать:
# - /var/lib/docker/containers:/var/lib/docker/containers:ro
# - /var/run/docker.sock:/var/run/docker.sock
networks: [elastic]
depends_on: [elasticsearch]
restart: unless-stopped

nginx-test:
  image: nginx:alpine
  container_name: nginx-test
  ports:
    - "8080:80"
  networks: [elastic]
  logging:
    driver: "json-file"
    options:
      max-size: "10m"
      max-file: "3"
  volumes:
    - ./nginx.conf:/etc/nginx.conf:ro
    - ./logs/nginx:/var/log/nginx

volumes:
  es_data:
    driver: local

networks:
  elastic:
    driver: bridge

```

Все контейнеры были запущены

```

stanik@archlinux::/home/stanik/programmer/++Programmer/5_1/BIG_DATA/lab/4/elk-logs > sudo docker compose up -d
[+] Running 5/5
✓ Network elk-logs_elastic   Created                                         0.0s
✓ Container elasticsearch     Started                                         0.3s
✓ Container nginx-test       Started                                         0.3s
✓ Container kibana           Started                                         0.4s
✓ Container filebeat         Started                                         0.4s
stanik@archlinux::/home/stanik/programmer/++Programmer/5_1/BIG_DATA/lab/4/elk-logs > █

```

и успешно функционируют

```

stanik@archlinux::/home/stanik/programmer/++Programmer/5_1/BIG_DATA/lab/4/elk-logs > sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                               NAMES
9e859849a99e        kibana:8.5.0      "/bin/tini -- /user/l..."   16 seconds ago   Up 15 seconds   0.0.0.0:5601->5601/tcp, [::]:5601->5601/tcp   kibana
13616fff47764       elastic/filebeat:8.11.0   "/usr/bin/tini -- /u..."   16 seconds ago   Up 15 seconds   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   filebeat
5a1022994037        nginx:alpine        "/docker-entrypoint..."   16 seconds ago   Up 16 seconds   0.0.0.0:9200->9200/tcp, [::]:9200->9200/tcp, 9300/tcp   nginx-test
f6d24f97daa        elasticsearch:8.5.0    "/bin/tini -- /user/l..."   16 seconds ago   Up 16 seconds   0.0.0.0:9200->9200/tcp, [::]:9200->9200/tcp, 9300/tcp   elasticsearch
stanik@archlinux::/home/stanik/programmer/++Programmer/5_1/BIG_DATA/lab/4/elk-logs > █

```

Следующий шаг: запустить скрипт для создания логов, посмотреть отправку и получение сообщений.

Bash-скрипт для записи логов посещения

```

for i in {1..10}; do
  curl http://localhost:8080/test-$i
  sleep 1
done

```

Web-UI Elastic (Kibana) после создания Data View:

The screenshot shows the Kibana Data Views interface. On the left, there's a navigation sidebar with sections like Management, Ingest, Data, Alerts and Insights, Kibana, Stack, and a Data Views section which is currently selected. The main area is titled "Data Views" and contains a search bar and a table. The table has columns for "Name" (with "filebeat-logs" listed), "Spaces" (with a small blue icon), and "Actions" (with a trash bin icon). Below the table, it says "Rows per page: 10". At the top right, there's a blue button labeled "Create data view".

Web-UI Elastic (Kibana) логи созданного Data View:

The screenshot shows the Kibana Discover interface. At the top, there's a search bar with the query "message: \"GET\"". Below it, there's a histogram showing "20 hits" over time from Oct 23, 2025 @ 15:09:00.000 to Oct 23, 2025 @ 15:09:30.000. The histogram has a single teal bar between 15:09:05 and 15:09:10. The main area displays a table of log entries. The columns are "Documents", "Field statistics", and "BETA". The table lists 20 hits, each with a timestamp (e.g., "Oct 23, 2025 @ 15:09:03.668") and a message. The messages all start with "message: \"GET\"". There are filters on the left for fields like "@_id", "@_index", "@score", and "@timestamp". A sidebar on the left shows available fields like "@_id", "@_index", "@score", and "@timestamp". At the bottom, there's a "Rows per page" dropdown set to 100.

Как видно по логам, запущенный скрипт посыпал GET-запросы по адресу [«http://localhost:8080/test-\\$i»](http://localhost:8080/test-$i), где i от 0 до 10. Но так как файлов test-1,2,3,... не существует, в логах записана лишь неудачная попытка обращения туда (порт 8080 — это порт nginx).

Логи контейнера filebeat:

Логи контейнера elasticsearch:

3. Вывод

В ходе выполнения лабораторной работы был настроен конфиг docker-compose.yml с целью получения логов из filebeat в WebUI elastic.

Приложение 1. docker-compose.yml

```
services:
  elasticsearch:
    image: elasticsearch:8.5.0
    container_name: elasticsearch
    environment:
      - discovery.type=single-node
      - ES_JAVA_OPTS=-Xms512m -Xmx512m
      - xpack.security.enabled=false
      - cluster.routing.allocation.disk.threshold_enabled=false
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - es_data:/usr/share/elasticsearch/data
    ports:
      - "9200:9200"
    networks: [elastic]

  kibana:
    image: kibana:8.5.0
    container_name: kibana
    environment:
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
      - ELASTICSEARCH_REQUESTTIMEOUT=120000
    ports:
      - "5601:5601"
    networks: [elastic]
    depends_on: [elasticsearch]

  filebeat:
    image: elastic/filebeat:8.11.0
    container_name: filebeat
    user: "0"                                     # чтобы точно было право читать файлы
    environment:
      - STRICT_PERMS=false
    volumes:
      - ./filebeat.yml:/usr/share/filebeat/filebeat.yml:ro
      - ./logs:/logs:ro                           # <-- монтируем папку с 1.log
    # Если не читаем логи контейнеров – эти два монтирования можно убрать:
    # - /var/lib/docker/containers:/var/lib/docker/containers:ro
    # - /var/run/docker.sock:/var/run/docker.sock
    networks: [elastic]
    depends_on: [elasticsearch]
    restart: unless-stopped

  nginx-test:
    image: nginx:alpine
    container_name: nginx-test
    ports:
      - "8080:80"
    networks: [elastic]
    logging:
      driver: "json-file"
```

```
options:
  max-size: "10m"
  max-file: "3"
volumes:
  - ./nginx.conf:/etc/nginx.conf:ro
  - ./logs/nginx:/var/log/nginx

volumes:
  es_data:
    driver: local

networks:
  elastic:
    driver: bridge
```

Приложение 2. Скрипт bash для генерации логов GET

```
for i in {1..10}; do
  curl http://localhost:8080/test-$i
  sleep 1
done
```