Devops-IA-1

Devops Tool: Fluentd

Name: Vihan Ajay Kumbhare

Roll No.: 16010122269

Class: COMPS-C2

What is Fluentd?

- Fluentd is an open-source data collector that helps you collect, transform, and ship logs/data from different sources to different destinations.
- It works as a unified logging layer, meaning it can take logs from apps, servers, containers, etc., and then forward them to places like Elasticsearch, Kafka, S3, Datadog, Splunk, CloudWatch, or just stdout.

Think of it like a central hub for logs.

Is Fluentd a DevOps Tool?

Yes – Fluentd is widely used in DevOps, SRE, and observability setups.

- In DevOps, one big challenge is log management. Applications, microservices, and containers generate tons of logs.
- Fluentd helps by collecting, filtering, and routing logs in a flexible way so that monitoring and alerting tools (like ELK, Prometheus, Grafana Loki, etc.) can use them.

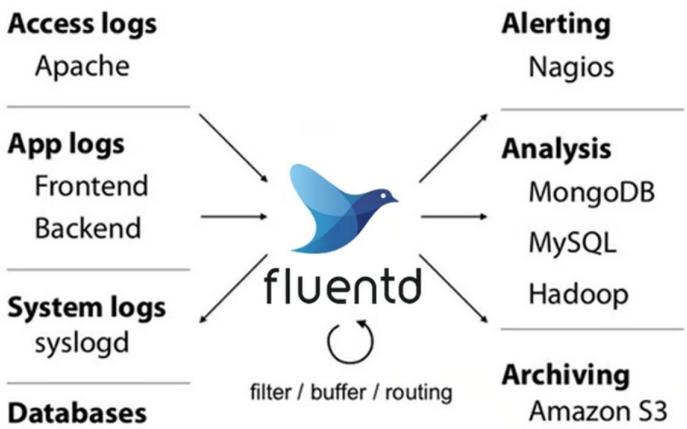
So, it's an important part of the DevOps logging & monitoring ecosystem.

Common Use Cases

Here's what you can use Fluentd for (like your example):

1. **Centralized Logging**
    - Collect logs from multiple apps/containers.
    - Store them in Elasticsearch → view with Kibana dashboards.
2. **Log Forwarding in Docker/Kubernetes**
    - Collect logs from pods/containers.
    - Send them to **CloudWatch, Loki, Elasticsearch, or Splunk**.

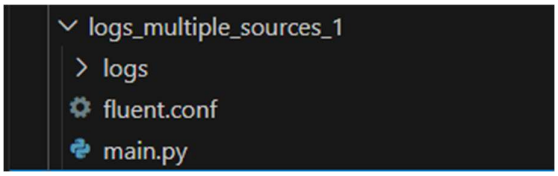
3. **Log Filtering & Transformation**
   - Mask sensitive data (passwords, tokens).
   - Reformat logs before sending them to a monitoring system.
4. **Monitoring & Alerting Support**
   - Pipe structured logs into monitoring tools.
   - Makes troubleshooting production issues easier.
5. **Data Routing**
   - Route logs to multiple places (for example → local file + S3 + Elasticsearch at the same time).



Getting the Fluentd Image from Docker

docker pull fluent/fluentd:v1.17-1

1. Collecting Logs from multiple sources



fluent.conf

```
<source>
  @type tail
  path /fluentd/log/app1.log
  pos_file /fluentd/log/app1.pos
  tag app1.log
  format none
</source>

<match **>
  @type stdout
</match>
```

main.py

```
import time
import random
import os

# Log file path
log_file = "./logs/app1.log"
os.makedirs("logs", exist_ok=True)

# Sample random English sentences
sentences = [
    "The quick brown fox jumps over the lazy dog.",
    "I love programming in Python.",
    "Fluentd makes log management easy.",
    "Docker containers simplify deployment.",
    "Learning new things every day keeps you sharp.",
    "Artificial Intelligence is the future of technology.",
    "Data science is both challenging and rewarding.",
    "Always keep your code clean and readable.",
    "Debugging can sometimes be fun.",
    "Consistency is key to mastering any skill.",
    "Reading books expands your knowledge.",
```

```
    "Writing tests improves software quality.",
    "Practice makes perfect.",
    "Collaboration leads to better solutions.",
    "Innovation drives progress.",
    "Automation saves time and reduces errors.",
    "Understanding algorithms is essential.",
    "Stay curious and keep exploring.",
    "Technology changes rapidly, adapt quickly.",
    "Good communication improves teamwork."
]

# Generate 20 log messages
for i in range(20):
    # Pick a random sentence
    message = random.choice(sentences)

    # Print to console
    print(message)

    # Append to log file
    with open(log_file, "a") as f:
        f.write(message + "\n")

    # Wait randomly between 5 and 10 seconds
    time.sleep(random.randint(5, 10))
```

Output

```
PS C:\Users\Lenovo\Devops-IA-1\Devops-IA-1\logs_multiple_sources_1> python main.py
Learning new things every day keeps you sharp.
Artificial Intelligence is the future of technology.
```

```
PS C:\Users\Lenovo\Devops-IA-1\devops-IA-1\logs_multiple_sources_1> docker run -it --rm -v "${PWD}\flue
nt.conf:/fluentd/etc/fluent.conf" -v "${PWD}\logs:/fluentd/log" fluent/fluentd:v1.17-1
2025-09-25 03:39:04 +0000 [info]: init supervisor logger path=nil rotate_age=nil rotate_size=nil
2025-09-25 03:39:04 +0000 [info]: parsing config file is succeeded path="/fluentd/etc/fluent.conf"
2025-09-25 03:39:04 +0000 [info]: gem 'fluentd' version '1.17.1'
2025-09-25 03:39:04 +0000 [warn]: define <match fluent.**> to capture fluentd logs in top level is depr
ecated. Use <label @FLUENT_LOG> instead
2025-09-25 03:39:04 +0000 [info]: using configuration file: <ROOT>
  <source>
    @type tail
    path "/fluentd/log/app1.log"
    pos_file "/fluentd/log/app1.pos"
    tag "app1.log"
    format none
    <parse>
      @type none
      unmatched_lines
    </parse>
  </source>
  <match **>
    @type stdout
  </match>
</ROOT>
2025-09-25 03:39:04 +0000 [info]: starting fluentd-1.17.1 pid=7 ruby="3.2.6"
2025-09-25 03:39:04 +0000 [info]: spawn command to main:  cmdline=["/usr/bin/ruby", "-Eascii-8bit:ascii
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

2025-09-25 03:39:05 +0000 [info]: adding match pattern="**" type="stdout"
2025-09-25 03:39:05 +0000 [info]: adding source type="tail"
2025-09-25 03:39:05 +0000 [warn]: #0 define <match fluent.**> to capture fluentd logs in top level is d
eprecated. Use <label @FLUENT_LOG> instead
2025-09-25 03:39:05 +0000 [info]: #0 starting fluentd worker pid=16 ppid=7 worker=0
2025-09-25 03:39:05 +0000 [info]: #0 following tail of /fluentd/log/app1.log
2025-09-25 03:39:05 +0000 [info]: #0 fluentd worker is now running worker=0
2025-09-25 03:39:05.113601651 +0000 fluent.info: {"pid":16,"ppid":7,"worker":0,"message":"starting flue
ntd worker pid=16 ppid=7 worker=0"}
2025-09-25 03:39:05.131279441 +0000 fluent.info: {"message":"following tail of /fluentd/log/app1.log"}
2025-09-25 03:39:05.133222782 +0000 fluent.info: {"worker":0,"message":"fluentd worker is now running w
orker=0"}
2025-09-25 03:39:10.125942055 +0000 app1.log: {"message":"Learning new things every day keeps you sharp
."}
2025-09-25 03:39:16.133549616 +0000 app1.log: {"message":"Artificial Intelligence is the future of tech
nology."}
2025-09-25 03:39:25.133389146 +0000 app1.log: {"message":"Writing tests improves software quality."}
2025-09-25 03:39:35.130856960 +0000 app1.log: {"message":"Data science is both challenging and rewardin
g."}
```

2. Filter and parse logs

```
Devops-IA-1
  filter_parse_2
    logs
    fluent.conf
    main.py
```

Fluent.conf

```
# Tail login log file
<source>
```

```
  @type tail
  path /fluentd/log/login.log
  pos_file /fluentd/log/login.pos
  tag login
  format json
</source>

# Filter: only SUCCESS logins
<filter login>
  @type grep
  <regexp>
    key status
    pattern ^SUCCESS$
  </regexp>
</filter>

# Add hostname field
<filter login>
  @type record_transformer
  <record>
    hostname "#{Socket.gethostname}"
  </record>
</filter>

# Output to console
<match login>
  @type stdout
</match>
```

main.py

```
import json
import os
from datetime import datetime

# Ensure logs folder exists
os.makedirs("logs", exist_ok=True)

# Predefined users
users = {
    "user1@example.com": "password123",
    "user2@example.com": "mypassword",
    "admin@example.com": "admin123"
}

# Ask user input
```

```python
email = input("Enter your email: ").strip()
password = input("Enter your password: ").strip()

# Verify credentials
status = "SUCCESS" if email in users and users[email] == password else
"FAILURE"
message = "Logged in successfully" if status == "SUCCESS" else "Failed to
login"

# Print to console
print(f"{message} for {email}")

# Create a structured log entry
log_entry = {
    "time": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    "email": email,
    "status": status,
    "message": message
}

# Append log entry as JSON to file
log_file = "./logs/login.log"
with open(log_file, "a") as f:
    f.write(json.dumps(log_entry) + "\n")
```

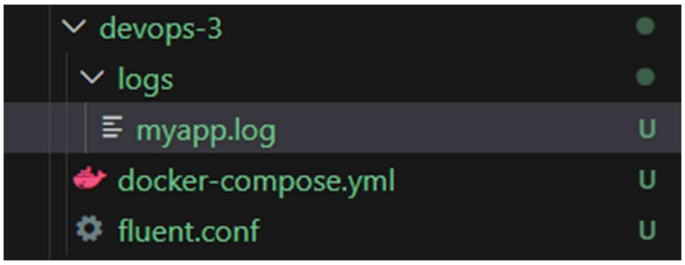
Output

```
PS C:\Users\Lenovo\Devops-IA-1\Devops-IA-1\filter_parse_2> python main.py
Enter your email: user1@example.com
Enter your password: password123
Logged in successfully for user1@example.com
PS C:\Users\Lenovo\Devops-IA-1\Devops-IA-1\filter_parse_2> python main.py
Enter your email: user@gmail.com
Enter your password: pass123
Failed to login for user@gmail.com
PS C:\Users\Lenovo\Devops-IA-1\Devops-IA-1\filter_parse_2>
```

```
PS C:\Users\Lenovo\Devops-IA-1\devops-IA-1\filter_parse_2> docker run -it --rm -v "${PWD}\fluent.conf:/
fluentd/etc/fluent.conf" -v "${PWD}\logs:/fluentd/log" fluent/fluentd:v1.17-1
2025-09-25 03:44:38 +0000 [info]: init supervisor logger path=nil rotate_age=nil rotate_size=nil
2025-09-25 03:44:38 +0000 [info]: parsing config file is succeeded path="/fluentd/etc/fluent.conf"
2025-09-25 03:44:38 +0000 [info]: gem 'fluentd' version '1.17.1'
2025-09-25 03:44:38 +0000 [info]: using configuration file: <ROOT>
  <source>
    @type tail
    path "/fluentd/log/login.log"
    pos_file "/fluentd/log/login.pos"
    tag "login"
    format json
    <parse>
      @type json
      unmatched_lines
    </parse>
  </source>
  <filter login>
    @type grep
    <regexp>
      key "status"
      pattern ^SUCCESS$
```

```
    @type record_transformer
    <record>
      hostname 3a928d3fdf92
    </record>
  </filter>
  <match login>
    @type stdout
  </match>
</ROOT>
2025-09-25 03:44:38 +0000 [info]: starting fluentd-1.17.1 pid=7 ruby="3.2.6"
2025-09-25 03:44:38 +0000 [info]: spawn command to main:  cmdline=["/usr/bin/ruby", "-Eascii-8bit:ascii
-8bit", "/usr/bin/fluentd", "--config", "/fluentd/etc/fluent.conf", "--plugin", "/fluentd/plugins", "--
under-supervisor"]
2025-09-25 03:44:39 +0000 [info]: #0 init worker0 logger path=nil rotate_age=nil rotate_size=nil
2025-09-25 03:44:39 +0000 [info]: adding filter pattern="login" type="grep"
2025-09-25 03:44:39 +0000 [info]: adding filter pattern="login" type="record_transformer"
2025-09-25 03:44:39 +0000 [info]: adding match pattern="login" type="stdout"
2025-09-25 03:44:39 +0000 [info]: adding source type="tail"
2025-09-25 03:44:39 +0000 [info]: #0 starting fluentd worker pid=16 ppid=7 worker=0
2025-09-25 03:44:39 +0000 [info]: #0 following tail of /fluentd/log/login.log
2025-09-25 03:44:39 +0000 [info]: #0 fluentd worker is now running worker=0
2025-09-25 03:44:49 +0000 [info]: #0 disable filter chain optimization because [Fluent::Plugin::RecordT
ransformerFilter] uses `#filter_stream` method.
1970-01-01 00:33:45.000000000 +0000 login: {"email":"user1@example.com","status":"SUCCESS","message":"L
ogged in successfully","hostname":"3a928d3fdf92"}
```

3. Send Logs to Elasticsearch (ELK Stack)



docker-compose.yml

```yaml
version: "3"
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:8.15.0
    container_name: elasticsearch
    environment:
      - discovery.type=single-node
      - xpack.security.enabled=false # disable auth for testing
    ports:
      - "9200:9200"

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

  fluentd:
    image: fluent/fluentd:v1.16-1
    container_name: fluentd
    volumes:
      - ./fluent.conf:/fluentd/etc/fluent.conf
      - ./logs:/var/log # directory with your logs
    depends_on:
      - elasticsearch
    ports:
      - "24224:24224"
      - "24224:24224/udp"
```

Fluent.conf

```
<source>
  @type tail
  path /var/log/myapp.log
  pos_file /fluentd/log/myapp.pos
  tag myapp.log
  format none
</source>

<match myapp.log>
  @type elasticsearch
```

```
  host elasticsearch    # service name if using Docker
  port 9200
  logstash_format true
  include_tag_key true
  type_name _doc
  flush_interval 5s
</match>
```

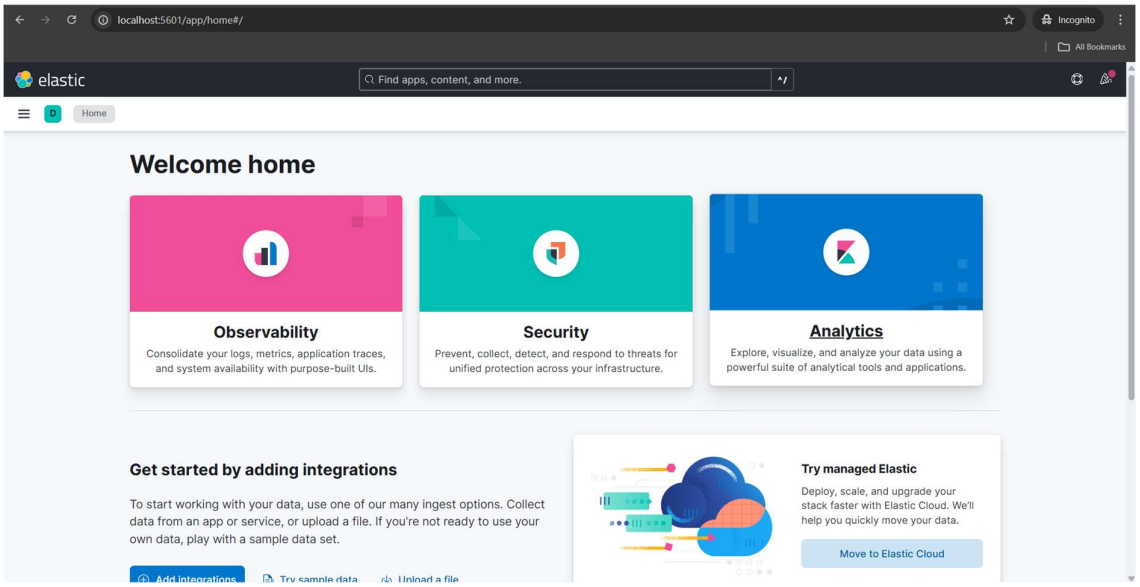
index.json

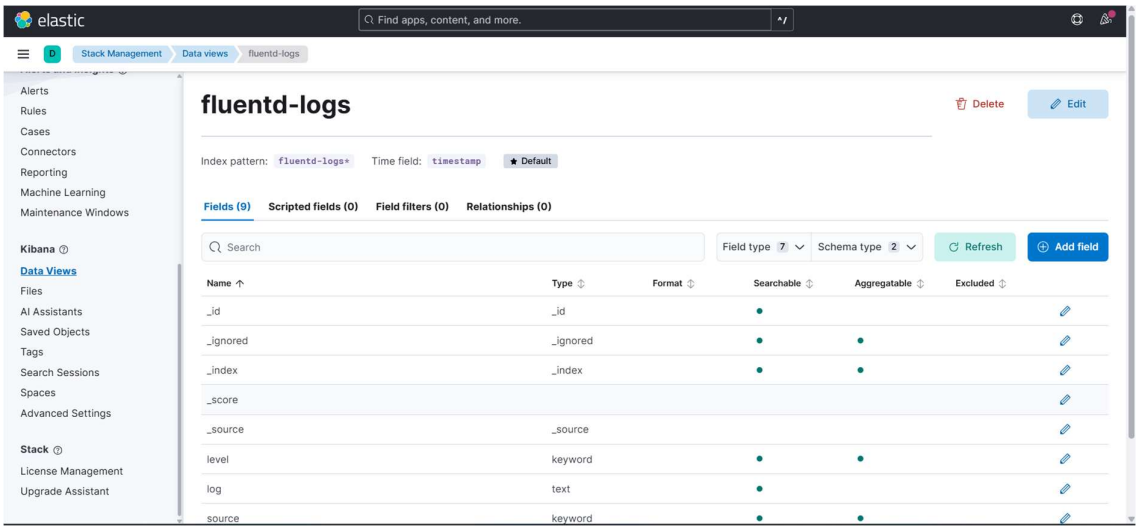
```
{
  "settings": {
    "number_of_shards": 1,
    "number_of_replicas": 0
  },
  "mappings": {
    "properties": {
      "timestamp": { "type": "date" },
      "log": { "type": "text" },
      "level": { "type": "keyword" },
      "source": { "type": "keyword" }
    }
  }
}
```

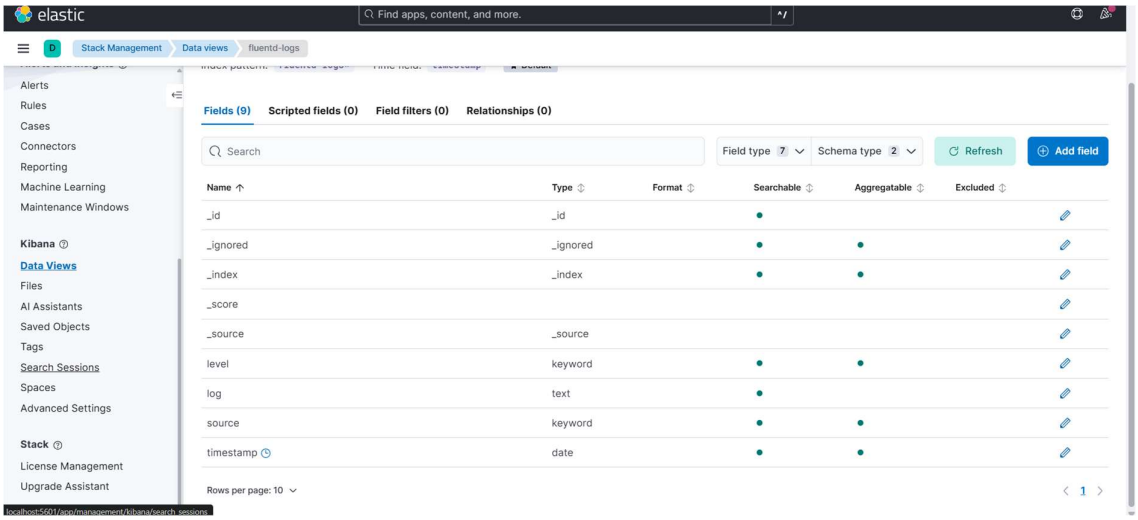
Output

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:32.138+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:32.151+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:32.179+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:33.142+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:33.179+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:35.192+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:35.199+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:39.410+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
kibana         | [2025-09-25T04:09:39.511+00:00][INFO ][plugins.fleet.fleet_authz_router] Kibana securi
ty must be enabled to use Fleet
Gracefully stopping... (press Ctrl+C again to force)
[+] Stopping 3/3
 ✓ Container fluentd        Stopped                                    0.0s
 ✓ Container kibana         Stopped                                    1.3s
 ✓ Container elasticsearch  Stopped                                    3.3s
exit status 130
PS C:\Users\Lenovo\Devops-IA-1\devops-3>
```




elastic
Find apps, content, and more.
Stack Management
Data views
fluentd-logs
Alerts
Rules
Cases
Connectors
Reporting
Machine Learning
Maintenance Windows
Kibana
Data Views
Files
AI Assistants
Saved Objects
Tags
Search Sessions
Spaces
Advanced Settings
Stack
License Management
Upgrade Assistant
fluentd-logs
Delete
Edit
Index pattern: fluentd-logs*
Time field: timestamp
Default
Fields (9)
Scripted fields (0)
Field filters (0)
Relationships (0)
Search
Field type 7
Schema type 2
Refresh
Add field
Name
Type
Format
Searchable
Aggregatable
Excluded
_id
_ignored
_index
_score
_source
level
log
source


elastic
Find apps, content, and more.
Stack Management
Data views
fluentd-logs
Alerts
Rules
Cases
Connectors
Reporting
Machine Learning
Maintenance Windows
Kibana
Data Views
Files
AI Assistants
Saved Objects
Tags
Search Sessions
Spaces
Advanced Settings
Stack
License Management
Upgrade Assistant
Fields (9)
Scripted fields (0)
Field filters (0)
Relationships (0)
Search
Field type 7
Schema type 2
Refresh
Add field
Name
Type
Format
Searchable
Aggregatable
Excluded
_id
_ignored
_index
_score
_source
level
log
source
timestamp
date
Rows per page: 10
1