UNVEILING THE POWER OF OBSERVABILITY USING FLUENT BIT IN AWS EKS ENVIRONMENT.

DHARMENDRA AHUJA
DEVOPS LEAD
IBM

May 24 2025

As cloud-native applications scale across Kubernetes clusters, gaining deep visibility into your workloads becomes both mission-critical and increasingly complex. In this session, we'll unveil how to harness the true power of observability in your AWS EKS environment using Fluent Bit—the blazing-fast, lightweight log processor built for modern infrastructure.

You'll learn how Fluent Bit can efficiently collect, parse, filter, and route logs from your containers, enriching them with Kubernetes metadata and streaming them to destinations like Amazon CloudWatch, OpenSearch, or any observability backend of your choice. We'll break down real-world use cases, show best practices for deploying Fluent Bit as a Daemon Set in EKS, and demonstrate how to make sense of the endless flow of logs that Kubernetes generates.

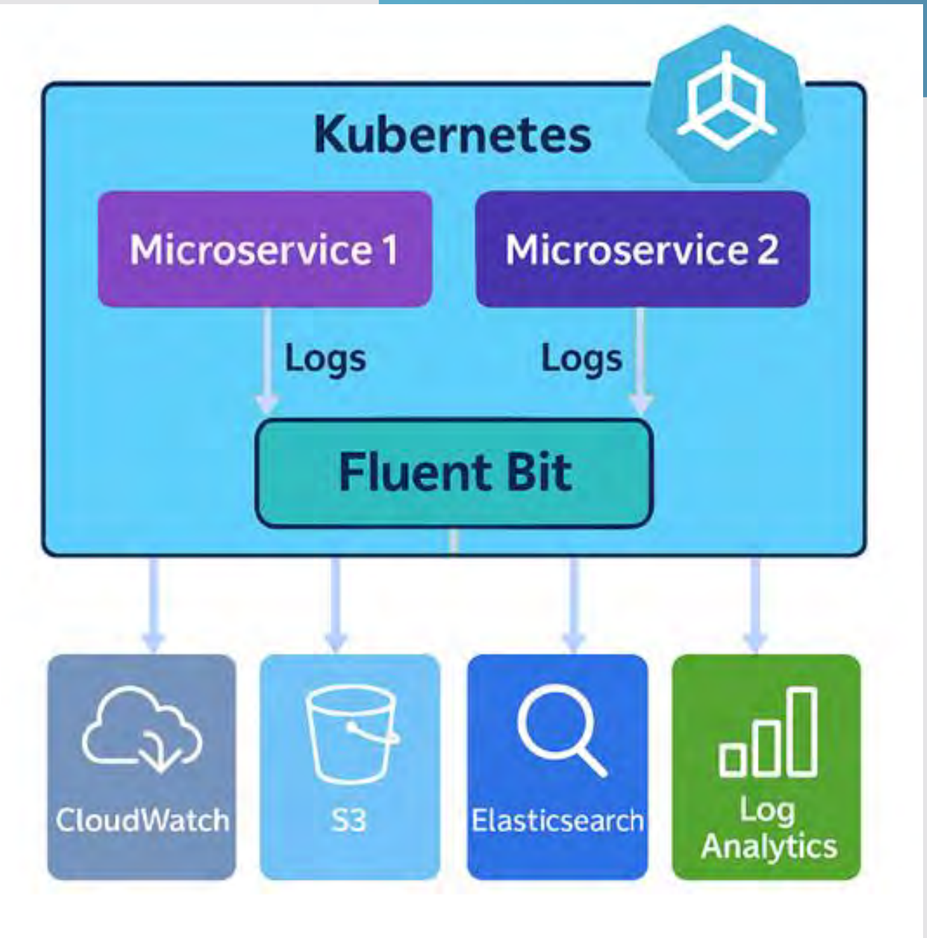
Introduction to Observability

- Observability = Ability to measure internal states by examining outputs
- Key Pillars: Logs, Metrics, Traces
- Importance in modern, cloud-native environments
- Why Kubernetes/EKS needs observability
- EKS runs complex, distributed containerized workloads.
- Observability helps you **track performance, health, and usage patterns** across services, nodes, and pods.
- It provides **visibility into control plane and data plane events.**
- Detect **pod crashes, OOM kills,** or **node pressure**.
- Monitor **CPU, memory, disk, and network** usage.

Observability in EKS

- EKS- Fully managed Kubernetes service by AWS. It Supports native tools for monitoring and logging
- Logs: Application, system, and Kubernetes logs
- Metrics: CPU, memory, custom metrics
- Traces: Request flows and performance tracing
- Native support for Fluent Bit, Prometheus, CloudWatch, X-Ray

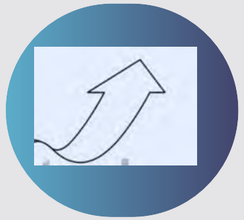
Introduction to Fluent Bit



Fluent Bit is a lightweight log processor and forwarder that allows you to collect data and logs from different sources, enrich them with filters and send them to multiple destinations like CloudWatch, Kinesis Data Firehose, Kinesis Data Streams and Amazon OpenSearch Service. Fluent Bit can be used to ship logs to various destinations. However, in this presentation, we will see how it shipped to Cloud Watch

Fluent Bit serves as the critical collection layer performing several essential functions

1. Collection: Capturing logs from containers, host paths, and application endpoints



2. Enrichment: Augmenting log data with Kubernetes metadata custom tags, and contextual information

3. Transformation: Structuring, filtering, and masking sensitive information via parsers and Lua scripts

4. Intelligent Routing: Directing logs to appropriate destinations based on content, namespace, or other attributes

How Fluentbit works..?
Data Flow Summary:

1. **Log Sources:** Logs originate from different sources like containers (Docker, Kubernetes), log files, syslog, or systemd/journald.

2. **Input Plugins:** These plugins collect logs (e.g., tail for log files, systemd for systemd journal).

3. **Parsers:** Transform raw log lines into structured data (e.g., JSON parsing).

How Fluentbit works..?
Data Flow Summary:

4 **Filters:** Modify, enrich, or remove specific log data before sending it to the destination.

5 **Buffers:** Temporarily store logs in memory or disk in case of high throughput or network delay.

6 **Output Plugins:** Push processed logs to external systems like Elasticsearch, Amazon S3, Kafka, or another Fluentd instance.

Setup

- To set up Fluent Bit to collect logs from your containers, the IAM role that is attached to the cluster nodes must have sufficient permissions.
- In the following steps, we will set up Fluent Bit as a daemonSet to send logs to CloudWatch Logs
- We will use Helm chart to install the CloudWatch Agent and the Fluent-bit agent on an Amazon EKS cluster.

Setup (contd.)

- Open ID Connect (OIDC) provider needs to be created
- IAM Policy need to be created, and it has to be associated with the IAM role while creating service account
- Following dependencies to be installed on machine
  - eksctl
  - kubectl
  - Helm
  - Gitbash

Setup (contd.)

logger-server.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: logger-server
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: main
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

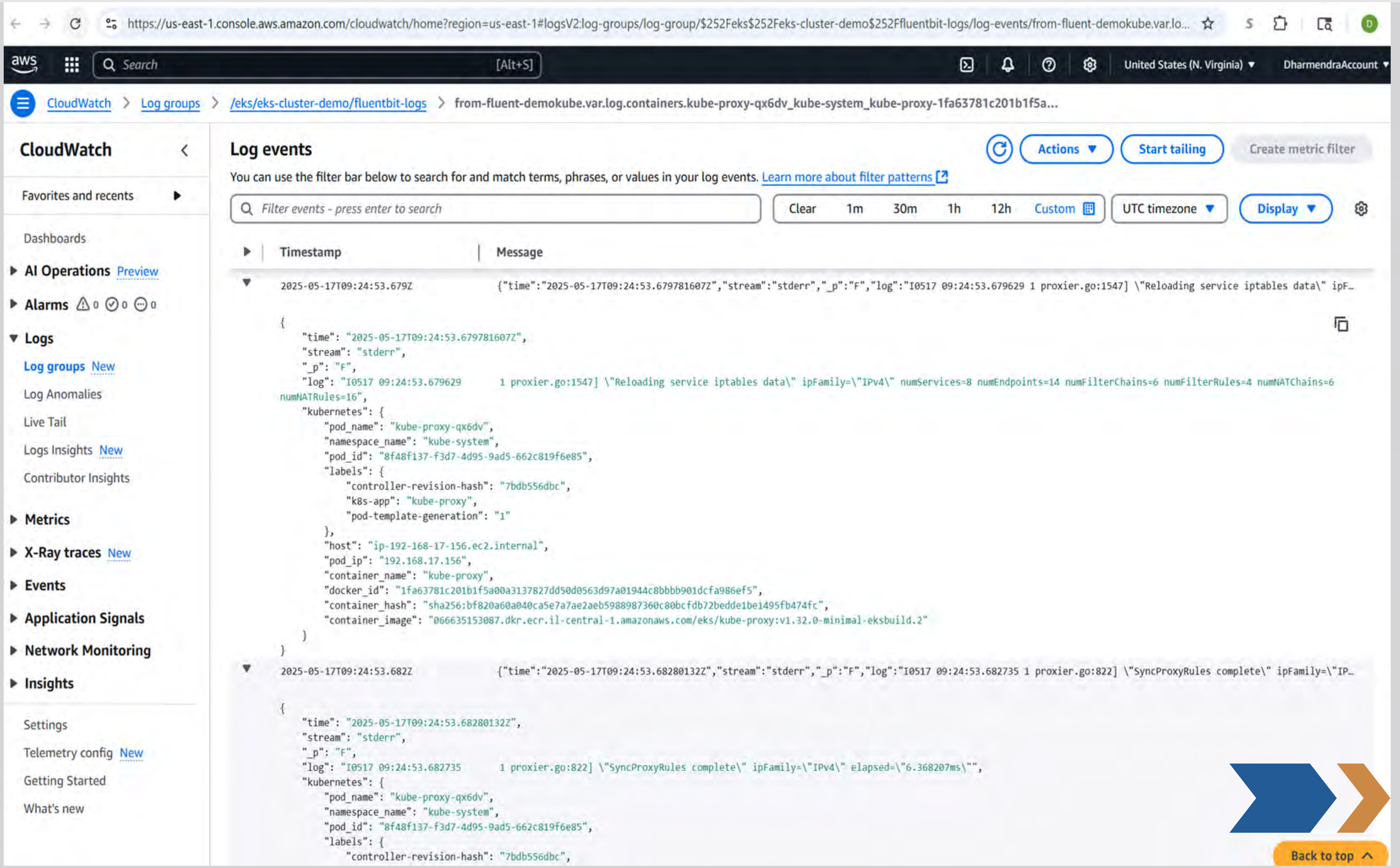
To cause the logger server to generate logs we forward the service to our local environment and use curl as the client to issue GET requests; in addition we watch the logs locally. Let's do that, using three terminals:

[terminal 1] forward the logger-server traffic locally:
kubectl -n demo port-forward svc/logger-server 8080:80

# [terminal 2] watch logs locally:
$ kubectl -n demo logs deploy/logger-server -f

# [terminal 3] generate HTTP traffic:
$ curl localhost:8080

Now, let's look at the cloud watch console to check if the logs are generated


https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups/log-group/$252Feks$252Feks-cluster-demo$252Ffluentbit-logs/log-events/from-fluent-demokube.var.lo...
aws
Search
[Alt+S]
United States (N. Virginia)
DharmendraAccount
CloudWatch
Log groups
/eks/eks-cluster-demo/fluentbit-logs
from-fluent-demokube.var.log.containers.kube-proxy-qx6dv_kube-system_kube-proxy-1fa63781c201b1f5a...
CloudWatch
Favorites and recents
Dashboards
AI Operations Preview
Alarms 0 0 0
Logs
Log groups New
Log Anomalies
Live Tail
Logs Insights New
Contributor Insights
Metrics
X-Ray traces New
Events
Application Signals
Network Monitoring
Insights
Settings
Telemetry config New
Getting Started
What's new
Log events
Actions
Start tailing
Create metric filter
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
Filter events - press enter to search
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Timestamp
Message
2025-05-17T09:24:53.679Z
{"time":"2025-05-17T09:24:53.679781607Z","stream":"stderr","_p":"F","log":"I0517 09:24:53.679629 1 proxier.go:1547] \"Reloading service iptables data\" ipF…
{
"time": "2025-05-17T09:24:53.679781607Z",
"stream": "stderr",
"_p": "F",
"log": "I0517 09:24:53.679629 1 proxier.go:1547] \"Reloading service iptables data\" ipFamily=\"IPv4\" numServices=8 numEndpoints=14 numFilterChains=6 numFilterRules=4 numNATChains=6 numNATRules=16",
"kubernetes": {
"pod_name": "kube-proxy-qx6dv",
"namespace_name": "kube-system",
"pod_id": "8f48f137-f3d7-4d95-9ad5-662c819f6e85",
"labels": {
"controller-revision-hash": "7bdb556dbc",
"k8s-app": "kube-proxy",
"pod-template-generation": "1"
},
"host": "ip-192-168-17-156.ec2.internal",
"pod_ip": "192.168.17.156",
"container_name": "kube-proxy",
"docker_id": "1fa63781c201b1f5a00a3137827dd50d0563d97a01944c8bbbb901dcfa986ef5",
"container_hash": "sha256:bf820a60a040ca5e7a7ae2aeb5988987360c80bcfdb72bedde1be1495fb474fc",
"container_image": "066635153087.dkr.ecr.il-central-1.amazonaws.com/eks/kube-proxy:v1.32.0-minimal-eksbuild.2"
}
}
2025-05-17T09:24:53.682Z
{"time":"2025-05-17T09:24:53.68280132Z","stream":"stderr","_p":"F","log":"I0517 09:24:53.682735 1 proxier.go:822] \"SyncProxyRules complete\" ipFamily=\"IP…
{
"time": "2025-05-17T09:24:53.68280132Z",
"stream": "stderr",
"_p": "F",
"log": "I0517 09:24:53.682735 1 proxier.go:822] \"SyncProxyRules complete\" ipFamily=\"IPv4\" elapsed=\"6.368207ms\"",
"kubernetes": {
"pod_name": "kube-proxy-qx6dv",
"namespace_name": "kube-system",
"pod_id": "8f48f137-f3d7-4d95-9ad5-662c819f6e85",
"labels": {
"controller-revision-hash": "7bdb556dbc",
Back to top

Demo

Real world examples:

1. Centralized Logging to CloudWatch
Company: Fintech Client

Challenge:

The bank runs multiple microservices on EKS and needed a unified log aggregation solution to comply with financial regulations requiring detailed audit trails and easy access to logs. Logs were previously scattered across pods and nodes.

Solution:

- Deployed Fluent Bit as a DaemonSet on EKS nodes.
- Configured Fluent Bit to tail container logs and enrich them with Kubernetes metadata.
- Forwarded all logs to Amazon CloudWatch Logs.
- Integrated CloudWatch Logs with AWS Lambda for alerting on anomalies.

Result:

- Centralizing log access improved troubleshooting efficiency and operational visibility.
- Automated alerts contributed to faster incident response.
- Compliance audits were streamlined with easy access to historical logs.

2. Streaming Logs to Amazon OpenSearch Company: SaaS Monitoring Provider

Challenge:

The company needed a scalable, low-latency solution to index logs from their Kubernetes-based monitoring agents deployed on EKS, allowing customers to query logs in near real-time.

Solution:

- Used Fluent Bit DaemonSet to collect logs from agent containers.
- Enriched logs with pod and namespace info using Fluent Bit filters.
- Streamed logs to Amazon OpenSearch Service clusters.
- Built Kibana dashboards for customers to analyze logs.

Result:

- Near real-time log availability improved monitoring and troubleshooting.
- Customers gained enhanced search and analytics capabilities.
- Operational overhead was reduced by leveraging managed OpenSearch service.

3. Forwarding Logs to Third-Party Services
Company: Online Gaming Platform

Challenge:

The gaming platform used Datadog for observability but needed an efficient way to send EKS container logs to Datadog without impacting cluster performance.

Solution:

- Deployed Fluent Bit with the http output plugin configured for Datadog's API.
- Applied filters to reduce unnecessary log volume in production.
- Used Kubernetes metadata enrichments for better log context.

Result:

- Reduced log ingestion costs through selective filtering.
- Improved debugging experience with enriched logs in Datadog dashboards.
- Seamless integration achieved without modifying application code.

4. Security & Compliance Monitoring
Company: Healthcare Provider

Challenge:

Maintaining HIPAA compliance required collecting and retaining audit logs from containerized workloads on EKS, including system and application logs.

Solution:

- Fluent Bit DaemonSet collects container logs and Linux audit logs via systemd input plugin.
- Logs were forwarded to a dedicated S3 bucket for long-term retention.
- Security team used Splunk to ingest from S3 for forensic analysis.

Result:

- Ensured compliance with regulatory requirements for log retention.
- Enabled more effective incident investigations.
- Reduced manual log management efforts.

5. Multi-Tenant Logging in EKS
Company: Managed Kubernetes Platform Provider

Challenge:

Platform served multiple customers (tenants) running their apps in isolated Kubernetes namespaces on a shared EKS cluster. Logs needed to be isolated and securely routed per tenant.

Solution:

- Fluent Bit deployed as DaemonSet with filters to add tenant labels based on namespaces.
- Configured routing rules to send logs to separate Grafana Loki tenants using the loki output plugin with tenant-specific URLs.
- Enforced access control so tenants could only view their own logs.

Result:

- Enabled multi-tenant log isolation without the need for separate clusters.
- Tenants gained self-service log access with ensured data privacy.
- Simplified platform operations with centralized log management.

References

- https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-setup-logs-FluentBit.html
- https://aws.amazon.com/blogs/containers/fluent-bit-for-amazon-eks-on-aws-fargate-is-here/.
- https://medium.com/@tolghn/advanced-logging-architecture-for-amazon-eks-with-fluent-bit-and-amazon-opensearch-service-9f2a80fff755.

THANK YOU

DHARMENDRA AHUJA

www.reallygreatsite.com