Fluent Bit Loki output plugin

- Configuration
- Usage examples
  - Tail Docker logs
    - Configuration file (Alternative to command line arguments)
  - Collect Docker events
    - Configuration file (Alternative to command line arguments)
  - Collect Kubernetes logs
- Next steps

Fluent Bit is a fast and lightweight logs and metrics processor and forwarder that can be configured with the Fluent-bit Loki output plugin to ship logs to Loki.

You can define which log files you want to collect using the `Tail` or `Stdin` data pipeline input. Additionally, Fluent Bit supports multiple `Filter` and `Parser` plugins (`Kubernetes`, `JSON`, etc.) to structure and alter log lines.

{{< admonition type="note" >}} There are two Fluent Bit plugins for Loki: the officially maintained plugin `loki` and the `grafana-loki` plugin. We recommend using the `loki` plugin described within this page as it's officially maintained by the Fluent Bit project.

For more information, see the Fluent Bit Loki output plugin documentation. Note that the `grafana-loki` plugin is no longer actively maintained. {{< /admonition >}}

Configuration

All configuration options for the Fluent Bit Loki output plugin are documented in the Fluent Bit Loki output plugin documentation.

Here is a generic example for connecting Fluent Bit to Loki hosted on Grafana Cloud:

```
    [OUTPUT]
        Name        loki
        Match       *
        Host        YourHostname.company.com
        port        443
        tls         on
        tls.verify  on
        http_user   XXX
        http_passwd XXX
```

Replace `Host`, `http_user`, and `http_passwd` with your Grafana Cloud Loki endpoint and credentials.

Usage examples

Here are some examples of how to use Fluent Bit to send logs to Loki.

Tail Docker logs

Here is an example to run Fluent Bit in a Docker container, collect Docker logs, and send them to a local Loki instance.

```
docker run -v /var/lib/docker/containers:/var/lib/docker/containers fluent/fluent-bit:latest
/fluent-bit/bin/fluent-bit -i tail -p Path="/var/lib/docker/containers/*/*.log" -p
Parser=docker -p Tag="docker.*"  -o loki -p host=loki -p port=3100 -p labels="agent=fluent-
bit,env=docker"
```

In this example, we are using the `tail` input plugin to collect Docker logs and the `loki` output plugin to send logs to Loki. Note it is recommended to use a configuration file to define the input and output plugins. The `-p` flag is used to pass configuration parameters to the plugins.

Configuration file (Alternative to command line arguments)

Create a configuration file `fluent-bit.conf` with the following content:

```
[INPUT]
    Name   tail
    Path   /var/lib/docker/containers/*/*.log
    Parser docker
    Tag    docker.*
```

```
[OUTPUT]
    Name   loki
    Match  *
    Host   loki
    Port   3100
    Labels agent=fluent-bit,env=docker
```

Run Fluent Bit with the configuration file:

```
docker run -v /var/lib/docker/containers:/var/lib/docker/containers -v $(pwd)/fluent-bit.conf:/fluent-bit/etc/fluent-bit.conf fluent/fluent-bit:latest /fluent-bit/bin/fluent-bit -c /fluent-bit/etc/fluent-bit.conf
```

Collect Docker events

Here is an example to run Fluent Bit in a Docker container, collect docker events, and send them to a local Loki instance.

```
docker run -v /var/run/docker.sock:/var/run/docker.sock fluent/fluent-bit:latest /fluent-bit/bin/fluent-bit -i docker_events -o loki -p host=loki -p port=3100 -p labels="agent=fluent-bit,env=docker"
```

In this example, we are using the `docker_events` input plugin to collect Docker events and the `loki` output plugin to send logs to Loki. Note it is recommended to use a configuration file to define the input and output plugins. The `-p` flag is used to pass configuration parameters to the plugins.

Configuration file (Alternative to command line arguments)

Create a configuration file `fluent-bit.conf` with the following content:

```
[INPUT]
    Name   docker_events

[OUTPUT]
    Name   loki
    Match  *
    Host   loki
    Port   3100
    Labels agent=fluent-bit,env=docker
```

Run Fluent Bit with the configuration file:

```
docker run -v /var/run/docker.sock:/var/run/docker.sock -v $(pwd)/fluent-bit.conf:/fluent-bit/etc/fluent-bit.conf fluent/fluent-bit:latest /fluent-bit/bin/fluent-bit -c /fluent-bit/etc/fluent-bit.conf
```

Collect Kubernetes logs

The recommended way to collect logs from Kubernetes with Fluent Bit is to use the Helm chart provided by the Fluent Bit project. The Helm chart is available at https://github.com/fluent/helm-charts.

Here is an example of how to deploy the Fluent Bit Helm chart to collect logs from Kubernetes and send them to Loki:

1. Add the Fluent Bit Helm repository:

   ```
   helm repo add fluent https://fluent.github.io/helm-charts
   ```

2. Create a `values.yaml` file with the following content:

   ```
   config:
       outputs: |
           [OUTPUT]
               Name        loki
               Match       *
               Host        YourHost.Company.net
               port        443
               tls         on
               tls.verify  on
               http_user   XXX
               http_passwd XXX
               Labels agent=fluent-bit
   ```

   Note we are only updating the `outputs` section of the Fluent Bit configuration. This is to replace the default output plugin with the Loki output plugin. If you need to update other parts of the Fluent Bit configuration refer to the Fluent Bit values file reference.

3. Deploy the Fluent Bit Helm chart:

   ```
   helm install fluent-bit fluent/fluent-bit -f values.yaml
   ```

Next steps

- Sending logs to Loki using Fluent Bit tutorial

□□□ #2
□ □□□ □□□ 2026-10-31 13:49:20 UTC
□ □□□ □□□ 2026-10-31 13:49:21 UTC