

Логирование в Kubernetes

1. Задача логирования в Kubernetes

- Логи контейнеров не должны теряться при рестартах Pod.
- Логи нужно централизовать.
- Формат и доставка логов должны быть стандартизированы.

2. Подходы к логированию

2.1. Логирование в stdout/stderr (основной подход)

Факты:

- kubelet пишет stdout/stderr контейнера в файлы `/var/log/containers/*.log`
- Работает из коробки.
- Удобно для простых сервисов.

Плюсы:

- Нет изменений в приложении.
- Совместимо со всеми лог-агентами.

Минусы:

- Нет структурирования.

2.2. Sidecar-контейнеры

Факты:

- В Pod добавляется контейнер, который читает лог-файлы приложения.
- Используется, если приложение не пишет в stdout.

Плюсы:

- Можно нормализовать и конвертировать логи.

Минусы:

- Усложняет Pod.
- Дополнительные ресурсы.

2.3. Агенты (Fluentd / Fluent Bit)

Факты:

- Устанавливаются как DaemonSet.
- Читают логи с `node` → отправляют в Loki/ES/S3.

Плюсы:

- Централизованный контроль.
- Фильтрация, парсинг, буферизация.

Минусы:

- Требует настройки и ресурсов на каждом узле.

3. Архитектура логирования с Loki

```
[ Pod ] --> stdout --> kubelet --> /var/log/containers/*.log  
          --> Fluent Bit --> Loki --> Grafana (чтение)
```

Ключевые компоненты:

- **Fluent Bit** — агент сбора логов
- **Loki** — хранилище
- **Grafana** — визуализация и поиск

4. Практика: настройка логирования в Loki

Шаг 1. Установить Loki stack через Helm

```
helm repo add grafana https://grafana.github.io/helm-charts  
helm repo update  
helm install loki grafana/loki-stack -n monitoring --create-namespace
```

Что разворачивается:

- Loki
- Promtail (агент по умолчанию)
- Grafana

Шаг 2. Проверить состояние

```
kubectl get pods -n monitoring
```

Ожидание:

- loki-...
- promtail-...
- grafana-...

Шаг 3. Проверить сбор логов

Promtail собирает файлы из `/var/log/containers`.

Проверка:

```
kubectl logs -n monitoring deployment/promtail
```

Шаг 4. Настроить Grafana для поиска логов

1. `kubectl port-forward -n monitoring service/grafana 3000:80`
2. Открыть `http://localhost:3000`
3. Перейти в *Explore* → *Loki*
4. Запрос:

```
{namespace="default"}
```

5. Итог

Что реализовано:

- Разобраны три подхода к логированию.
- Настроен сбор логов в Loki.
- Настроена визуализация через Grafana.

Где использовать:

- Любые кластеры Kubernetes (dev/prod).
- Можно интегрировать с GitOps и CI/CD.