## **Study: Collecting Argo CD Audit Logs**

### **1. Introduction**

### This document summarizes the study performed to enable the collection of **Argo CD audit logs** within the Kubernetes platform. The goal was to capture all user and API actions (application creation, synchronization, project and cluster management, deletions, etc.) in a centralized logging solution for audit and observability purposes.

### **2. Initial Approach: Fluent Bit Sidecar**

The first approach consisted of deploying **Fluent Bit** as a sidecar container next to the *argocd-server* pod.  
The idea was to:

* Redirect *argocd-server* logs to a local file inside the pod (*emptyDir* volume).
* Configure Fluent Bit to tail that file and forward the logs to a Location (NOT YET DEFINED).

However, this solution required **mounting hostPath volumes** ( */var/log/containers*) to access container logs at node level.  
Since our cluster nodes are **fully managed** (no privileged access, no direct hostPath mounts allowed), this implementation failed due to **missing permissions**.

A request for additional permissions was raised to the **Orchestrated Container Team** under the following ticket: XXXXX

### **3. Feedback from the Kubernetes Team**

After reviewing the request, the **Orchestrated Container Team** informed that:

* Fluent Bit is **not required**, since the platform already runs a **Filebeat + Metricbeat DaemonSet**.
* The existing **Filebeat DaemonSet** is centrally managed by the **SRE Team**, applied by **Kubernetes Team** and already has access to */var/log/containers* through hostPath mounts.
* Filebeat is configured to send all collected logs to the **central Elastic Stack (Elasticsearch + Kibana)**.

We confirmed with Chema that the **Argo CD logs are already visible in Kibana**.

Manifests SGithub URL: XXXXX

Kibana URL: XXXXX

### **4. Final Decision**

Given the existing setup, deploying an additional Fluent Bit component is unnecessary and redundant.  
The agreed solution is to **reuse the existing Filebeat DaemonSet** managed by the **Orchestrated Container** Team.

To specifically capture and structure Argo CD audit logs, the plan is to:

* Create a small **ConfigMap** containing an extra *filestream* input dedicated to Argo CD.
* Mount it into the existing Filebeat DaemonSet (under */usr/share/filebeat/inputs.d/argocd.yml*).
* Use existing outputs (Elasticsearch) and pipelines already configured by the **SRE Team**.
* Optionally route these logs to a dedicated index *argocd-audit-\**.

### **6. Proposed Configuration Addition**

Below is the configuration that can be added safely without altering the existing Filebeat setup:

*- type: filestream*

*id: argocd-server*

*paths:*

*- /var/log/containers/\*\_argocd\_argocd-server-\*.log*

*parsers:*

*- container: ~*

*fields:*

*service.name: argocd*

*service.component: server*

*event.dataset: argocd.audit*

*fields\_under\_root: true*

This configuration:

* Reads only the Argo CD Server logs from */var/log/containers/*.
* Adds metadata fields to identify audit logs.
* Leverages existing outputs (no additional credentials or pipelines required).

### **7. Future Steps**

1. Prepare ConfigMap and patch for Filebeat DaemonSet
2. Validate configuration in production (NOTE: here we don’t have this config in dev, it exists only in prod).
3. Confirm visibility of logs in Kibana
4. Define dedicated index pattern / dashboard in Kibana

### **References**

**GitHub Repository (filebeat manifests):** XXXXX

**Kibana Dashboard:** XXXXXX

**Ticket SRE team:** XXXXXX