

# Logging Storage

## 1. Loki Storage Backend

Loki stores logs in a backend system that is configured when Loki is deployed. It uses two main components for storage:

### a. Index Store

- Stores metadata (labels, timestamps, etc.) to allow quick search and filtering of logs.
- By default, Loki minimizes the amount of indexing (compared to systems like Elasticsearch) to optimize for cost and speed.

### b. Log Store (Chunks)

- Stores the actual log content in compressed chunks.
- The logs are stored in object storage, or local disk, depending on the configuration.

## Common Storage Options for Loki

Loki supports multiple storage backends for scalability and reliability:

### 1. Local Disk

- Logs are stored on the server's local filesystem.
- Best suited for small-scale setups or testing environments.

Example configuration:

```
storage_config:  
  boltdb_shipper:  
    active_index_directory: /tmp/loki/boltdb-shipper-active  
  filesystem:  
    directory: /tmp/loki/chunks
```

### 2. Object Storage (Cloud or On-Premise)

- Amazon S3, Google Cloud Storage (GCS), Azure Blob Storage, MinIO, etc.
- Preferred for large-scale production setups due to high durability and scalability.

Example configuration for S3:

```
storage_config:  
  aws:  
    s3: s3://<bucket-name>  
    region: us-west-2
```

### 3. Network Storage

- Shared network file systems like NFS.
- Useful for redundancy across multiple nodes.

### 4. DynamoDB (Indexing only)

- Loki can store its index in Amazon DynamoDB while storing chunks in S3.
- This is common in highly distributed systems.

## 2. Where Does Grafana Fit?

- Grafana itself doesn't store logs.
- It queries Loki, which retrieves logs from its storage backend and serves them to Grafana for visualization.

### How to Check or Configure Loki Storage

- Configuration files (typically loki-config.yaml) define the storage backend.
- Example for Kubernetes:
  - Loki configuration is usually passed as a ConfigMap.
  - Check the ConfigMap by running:

```
kubectl get configmap loki-config -n <namespace> -o yaml
```

---

Revision #1

Created 15 November 2024 05:38:10 by Ahmad

Updated 15 November 2024 05:48:48 by Ahmad