Introduction

Loki is a powerful and flexible open-source log aggregation system designed to be highly scalable and efficient. It is a part of the Grafana Labs ecosystem and is particularly well-suited for cloud-native environments.

Key Features

Log Aggregation: Loki is designed to collect, store, and query log data efficiently. It allows you to aggregate logs from various sources in one centralized location.

Highly Scalable: Loki scales horizontally, making it suitable for both small and large-scale deployments. It efficiently handles growing log volumes.

Prometheus Integration: Loki seamlessly integrates with Prometheus, a popular monitoring and alerting toolkit. This integration enhances the ability to correlate metrics and logs for better troubleshooting.

Cost-Effective: Loki follows a cost-effective storage model, utilizing object storage like Amazon S3 or Google Cloud Storage. This enables you to store large volumes of log data without incurring exorbitant costs.

Query Language: Loki uses a powerful label-based query language, inspired by PromQL. This allows for flexible and efficient log querying.

Loki is a horizontally-scalable, highly-available, multi-tenant log aggregation system built by Grafana Labs.

Loki server :serves as storage, storing the logs in a time series database, but it won’t index them. To visualize the logs, you need to extend Loki with Grafana in combination with LogQL

Loki agents

will be deployed as a DaemonSet, and they're in charge of collecting logs from various pods/containers of our nodes. Loki supports various types of agents, but the default one is called Promtail.

Promtail does the following actions:

* it discovers the targets having logs
* it attaches labels to log streams
* And it pushes the log stream to Loki

**Client configuration**. To specify how it connects to Loki.

Positioning. To make Promtail reliable in case it crashes and avoid duplicates.

Scrape config: That will specify each job that will be in charge of collecting the logs.

Relabel config: That will control what to ingest, what to drop, what type of metadata to attach to the log line.

Installation(https://grafana.com/docs/loki/latest/setup/install/local/)

To download the loki-linux-amd64.zip file

curl -O -L "https://github.com/grafana/loki/releases/download/v2.9.2/loki-linux-amd64.zip"

To unzip the file

unzip "loki-linux-amd64.zip"

**URL to download the configuration file that corresponds to the Loki version you aim to run.**

wget <https://raw.githubusercontent.com/grafana/loki/main/cmd/loki/loki-local-config.yaml>

Enter the following command to start Loki:

./loki-linux-amd64 -config.file=loki-local-config.yaml

For promtail also same like loki

To download the promtail-linux-amd64.zip file

curl -O -L <https://github.com/grafana/loki/releases/download/v2.9.2/promtail-linux-amd64.zip>

To unzip the file

unzip "promtail-linux-amd64.zip"

**URL to download the configuration file that corresponds to the Loki version you aim to run.**

wget <https://raw.githubusercontent.com/grafana/loki/main/clients/cmd/promtail/promtail-local-config.yaml>

Enter the following command to start

Sudo ./promtail-linux-amd64 -config.file=promtail-local-config.yaml

Start the Grafana : Systemctl start Grafana-services

Open :10.81.1.66(vm ip):3000 on browser

K8 :

sudo snap install helm –classic

helm repo add grafana <https://grafana.github.io/helm-charts>

helm repo update

helm search repo loki

helm show values grafana/loki-stack

helm show values grafana/loki-stack > values.yaml

helm install --values values.yaml loki grafana/loki-stack

loki-kudernetes$ kubectl port-forward pod/loki-grafana-6f87885599-srfc92d 3000:3000