How to efficiently manage logs in large-scale Kubernetes clusters

Open Source Observability Day 2024

Aliaksandr Valialkin, CTO at VictoriaMetrics

About me (valyala)

- I’m software engineer

About me (valyala)

- I’m software engineer
- I like writing fast code in Go

About me (valyala)

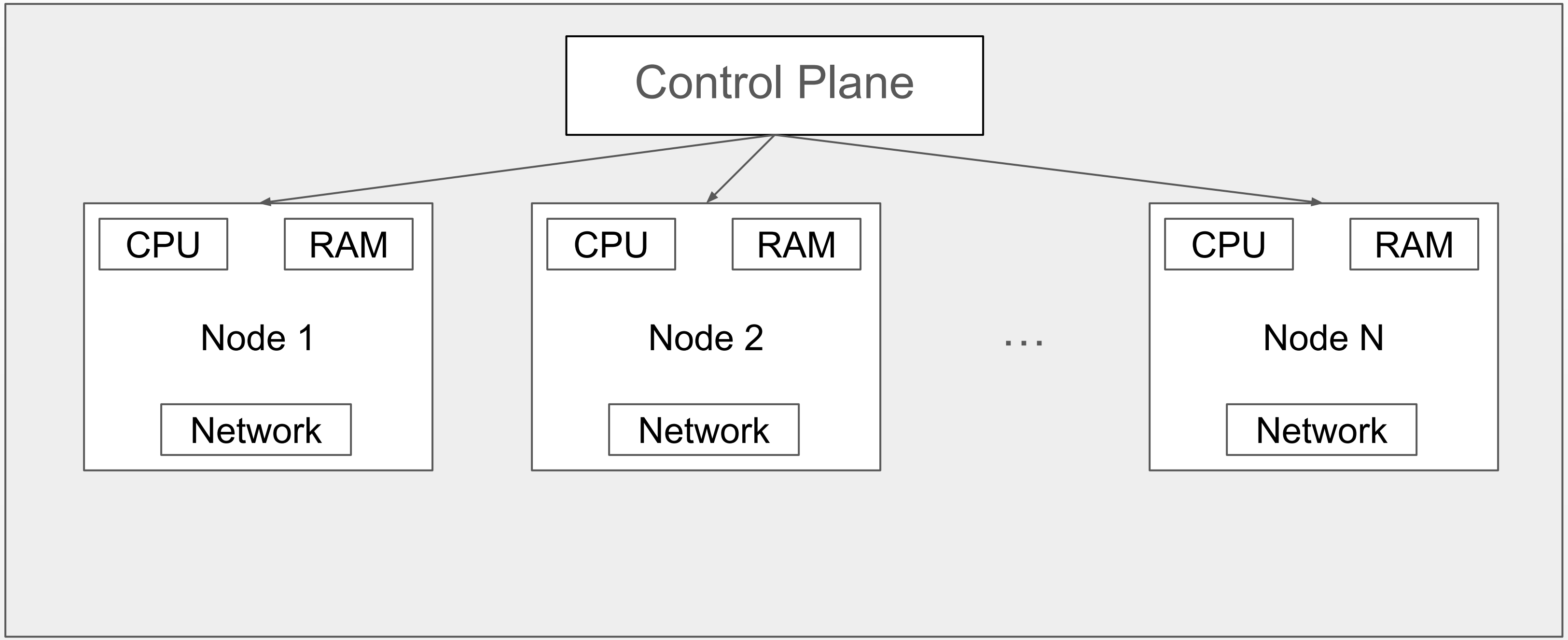
- I’m software engineer
- I like writing fast code in Go
- I work on specialized open-source databases
    - VictoriaMetrics - time series database

About me (valyala)

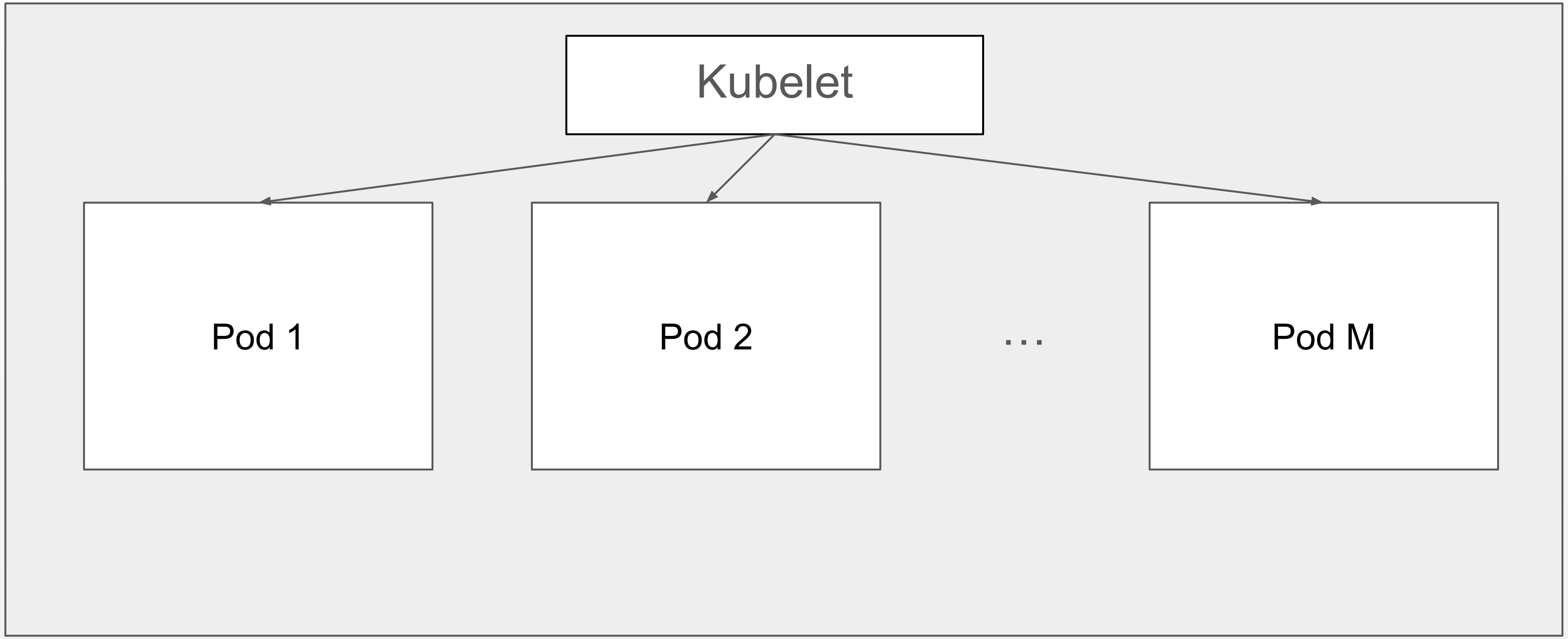
- I’m software engineer
- I like writing fast code in Go
- I work on specialized open-source databases
    - VictoriaMetrics - time series database
    - VictoriaLogs - database for logs

Kubernetes cluster

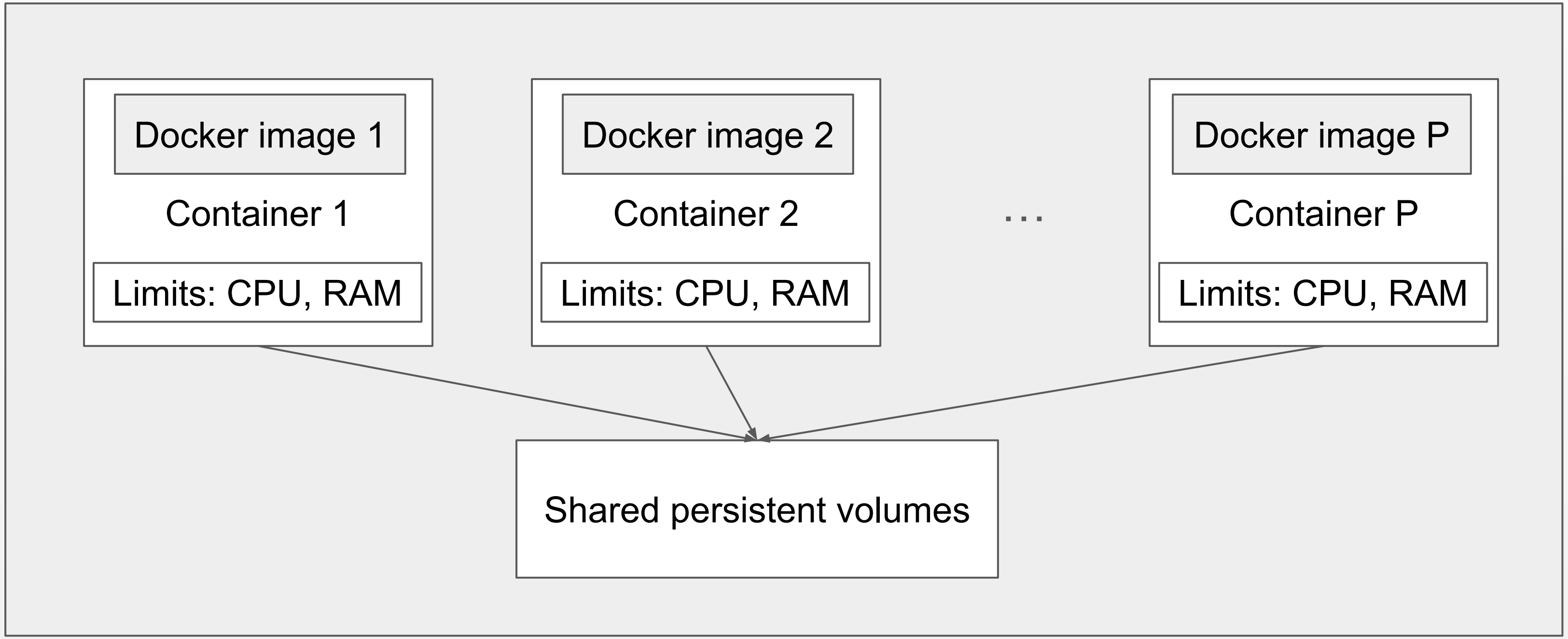
Kubernetes cluster



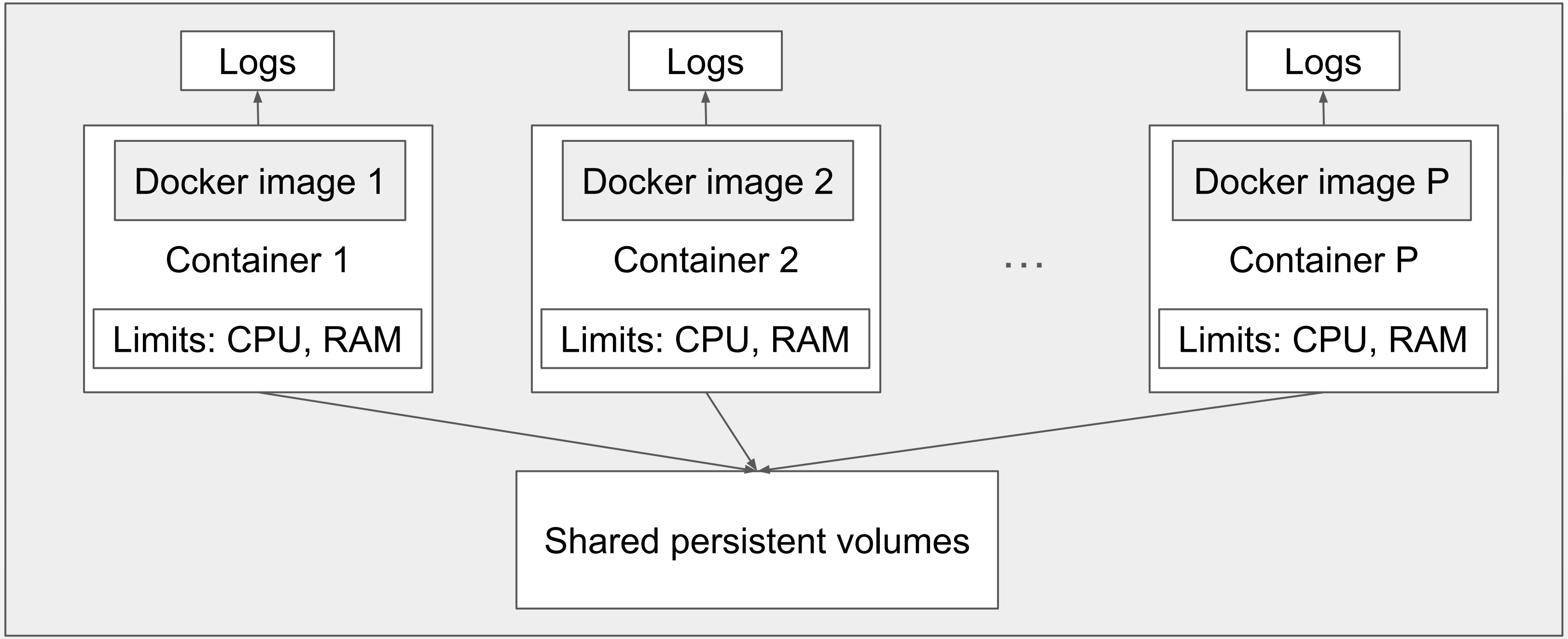
Kubernetes node



Kubernetes pod



Kubernetes logs



Kubernetes log sources

Kubernetes log sources

- Kubernetes containers

Kubernetes log sources

- Kubernetes containers
    - Control plane (apiserver, etcd, kube-scheduler, etc.)

Kubernetes log sources

- Kubernetes containers
  - Control plane (apiserver, etcd, kube-scheduler, etc.)
  - Node services (kubelet, kube-proxy, container runtime, etc.)

Kubernetes log sources

- Kubernetes containers
  - Control plane (apiserver, etcd, kube-scheduler, etc.)
  - Node services (kubelet, kube-proxy, container runtime, etc.)
- User containers

Kubernetes log sources

- Kubernetes containers
  - Control plane (apiserver, etcd, kube-scheduler, etc.)
  - Node services (kubelet, kube-proxy, container runtime, etc.)
- User containers

The most of Kubernetes logs are usually generated by containers deployed by users (aka microservices)

Kubernetes container logs

Kubernetes container logs: destination

- stdout / stderr

Kubernetes container logs: destination

- stdout / stderr
- custom files

Kubernetes container logs: destination

- stdout / stderr
- custom files
- external database for logs

Kubernetes container logs: destination

- stdout / stderr
- custom files
- external database for logs

Stdout / stderr is the standard destination for container logs in Kubernetes

Kubernetes container logs: lifecycle

- Kubernetes automatically collects stdout / stderr logs from every running container

Kubernetes container logs: lifecycle

- Kubernetes automatically collects stdout / stderr logs from every running container
- Per-container logs are stored into distinct files on the local Kubernetes node

Kubernetes container logs: lifecycle

- Kubernetes automatically collects stdout / stderr logs from every running container
- Per-container logs are stored into distinct files on the local Kubernetes node
- Every log file size is **limited with 10MB**. Older logs are automatically dropped

Kubernetes container logs: lifecycle

- Kubernetes automatically collects stdout / stderr logs from every running container
- Per-container logs are stored into distinct files on the local Kubernetes node
- Every log file size is **limited with 10MB**. Older logs are automatically dropped
- Logs for stopped containers are **eventually dropped**

Kubernetes container logs: lifecycle

- Kubernetes automatically collects stdout / stderr logs from every running container
- Per-container logs are stored into distinct files on the local Kubernetes node
- Every log file size is **limited with 10MB**. Older logs are automatically dropped
- Logs for stopped containers are **eventually dropped**
- Container logs can be inspected with **kubectl logs** command

Kubernetes container logs: lifecycle

- Kubernetes automatically collects stdout / stderr logs from every running container
- Per-container logs are stored into distinct files on the local Kubernetes node
- Every log file size is **limited with 10MB**. Older logs are automatically dropped
- Logs for stopped containers are **eventually dropped**
- Container logs can be inspected with **kubectl logs** command

```
kubectl logs pod_name -c container_name
```

“kubectl logs” features

- Scales to millions of containers

“kubectl logs” features

- Scales to millions of containers
- Supports “tail -f” functionality

“kubectl logs” features

- Scales to millions of containers
- Supports “tail -f” functionality
- Good integration with traditional Unix command-line tools (grep, head, tail, etc.)

“kubectl logs” issues

- Shows only **the last 10MB** of logs per container by default

“kubectl logs” issues

- Shows only **the last 10MB** of logs per container by default
- Logs for stopped containers are **lost eventually**

“kubectl logs” issues

- Shows only **the last 10MB** of logs per container by default
- Logs for stopped containers are **lost eventually**
- Doesn’t provide the ability to quickly search over large volume of logs across containers
    - Find all logs with **trace_id=XXXXXXXX**

“kubectl logs” issues

- Shows only **the last 10MB** of logs per container by default
- Logs for stopped containers are **lost eventually**
- Doesn’t provide the ability to quickly search over large volume of logs across containers
  - Find all logs with **trace_id=XXXXXXXX**
- Doesn’t provide tools for log analytics
  - Find top 5 containers with the highest volumes of logs

Large-scale logging solution for Kubernetes: requirements

Large-scale logging solution for Kubernetes: requirements

- To forward logs from all the containers to a centralized database for logs

Large-scale logging solution for Kubernetes: requirements

- To forward logs from all the containers to a centralized database for logs
  - Allows managing logs in a centralized manner

Large-scale logging solution for Kubernetes: requirements

- To forward logs from all the containers to a centralized database for logs
    - Allows managing logs in a centralized manner
    - Allows querying all the logs at once

Large-scale logging solution for Kubernetes: requirements

- To forward logs from all the containers to a centralized database for logs
    - Allows managing logs in a centralized manner
    - Allows querying all the logs at once
- Requirements for database for logs:

Large-scale logging solution for Kubernetes: requirements

- To forward logs from all the containers to a centralized database for logs
  - Allows managing logs in a centralized manner
  - Allows querying all the logs at once
- Requirements for database for logs:
  - Ability to **efficiently** accept and store huge log volumes

Large-scale logging solution for Kubernetes: requirements

- To forward logs from all the containers to a centralized database for logs
  - Allows managing logs in a centralized manner
  - Allows querying all the logs at once
- Requirements for database for logs:
  - Ability to **efficiently** accept and store huge log volumes
  - Fast full-text search over **terabytes** of ingested logs

Large-scale logging solution for Kubernetes: requirements

- To forward logs from all the containers to a centralized database for logs
  - Allows managing logs in a centralized manner
  - Allows querying all the logs at once
- Requirements for database for logs:
  - Ability to **efficiently** accept and store huge log volumes
  - Fast full-text search over **terabytes** of ingested logs
  - Fast analytics over the **billions** of ingested logs

Databases for Kubernetes logs

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
  - Works OK for small volumes of logs

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
  - Works OK for small volumes of logs
  - Scalability issues on large log volumes (needs a lot of disk space, disk IO, RAM and CPU)

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
  - Works OK for small volumes of logs
  - Scalability issues on large log volumes (needs a lot of disk space, disk IO, RAM and CPU)
  - Not-so-good usability for typical logging tasks

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
  - Works OK for small volumes of logs
  - Scalability issues on large log volumes (needs a lot of disk space, disk IO, RAM and CPU)
  - Not-so-good usability for typical logging tasks
- Analytical databases - ClickHouse
  - Scales great for large log volumes

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
  - Works OK for small volumes of logs
  - Scalability issues on large log volumes (needs a lot of disk space, disk IO, RAM and CPU)
  - Not-so-good usability for typical logging tasks
- Analytical databases - ClickHouse
  - Scales great for large log volumes
  - **Non-trivial** to configure properly

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
  - Works OK for small volumes of logs
  - Scalability issues on large log volumes (needs a lot of disk space, disk IO, RAM and CPU)
  - Not-so-good usability for typical logging tasks
- Analytical databases - ClickHouse
  - Scales great for large log volumes
  - **Non-trivial** to configure properly
  - Needs custom tools for simplifying logs’ ingestion and querying

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
  - Works OK for small volumes of logs
  - Scalability issues on large log volumes (needs a lot of disk space, disk IO, RAM and CPU)
  - Not-so-good usability for typical logging tasks
- Analytical databases - ClickHouse
  - Scales great for large log volumes
  - **Non-trivial** to configure properly
  - Needs custom tools for simplifying logs’ ingestion and querying
- Document indexing databases - Elasticsearch, OpenSearch
  - Fast search and analytics over logs

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
    - Works OK for small volumes of logs
    - Scalability issues on large log volumes (needs a lot of disk space, disk IO, RAM and CPU)
    - Not-so-good usability for typical logging tasks
- Analytical databases - ClickHouse
    - Scales great for large log volumes
    - **Non-trivial** to configure properly
    - Needs custom tools for simplifying logs’ ingestion and querying
- Document indexing databases - Elasticsearch, OpenSearch
    - Fast search and analytics over logs
    - Non-trivial setup and operation

Databases for Kubernetes logs

- Traditional databases - MySQL, PostgreSQL, etc.
  - Works OK for small volumes of logs
  - Scalability issues on large log volumes (needs a lot of disk space, disk IO, RAM and CPU)
  - Not-so-good usability for typical logging tasks
- Analytical databases - ClickHouse
  - Scales great for large log volumes
  - **Non-trivial** to configure properly
  - Needs custom tools for simplifying logs’ ingestion and querying
- Document indexing databases - Elasticsearch, OpenSearch
  - Fast search and analytics over logs
  - Non-trivial setup and operation
  - Needs **a ton of RAM** for large log volumes

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs
  - Slow search over large volumes of logs

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs
  - Slow search over large volumes of logs
  - Doesn’t support log high-cardinality log fields (user_id, trace_id, ip)

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs
  - Slow search over large volumes of logs
  - Doesn’t support log high-cardinality log fields (user_id, trace_id, ip)
  - **Hard to configure and operate properly**

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs
  - Slow search over large volumes of logs
  - Doesn’t support log high-cardinality log fields (user_id, trace_id, ip)
  - **Hard to configure and operate properly**
- VictoriaLogs
  - Trivial to setup and operate

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs
  - Slow search over large volumes of logs
  - Doesn’t support log high-cardinality log fields (user_id, trace_id, ip)
  - **Hard to configure and operate properly**
- VictoriaLogs
  - Trivial to setup and operate
  - Low disk space usage (**up to 15x less than Elasticsearch**)

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs
  - Slow search over large volumes of logs
  - Doesn’t support log high-cardinality log fields (user_id, trace_id, ip)
  - **Hard to configure and operate properly**
- VictoriaLogs
  - Trivial to setup and operate
  - Low disk space usage (**up to 15x less than Elasticsearch**)
  - Low RAM usage (**up to 30x less than Elasticsearch**)

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs
  - Slow search over large volumes of logs
  - Doesn’t support log high-cardinality log fields (user_id, trace_id, ip)
  - **Hard to configure and operate properly**
- VictoriaLogs
  - Trivial to setup and operate
  - Low disk space usage (**up to 15x less than Elasticsearch**)
  - Low RAM usage (**up to 30x less than Elasticsearch**)
  - Fast full-text search over large volumes of logs

Databases for Kubernetes logs: specialized databases

- Loki
  - Good compression for logs
  - Slow search over large volumes of logs
  - Doesn’t support log high-cardinality log fields (user_id, trace_id, ip)
  - **Hard to configure and operate properly**
- VictoriaLogs
  - Trivial to setup and operate
  - Low disk space usage (**up to 15x less than Elasticsearch**)
  - Low RAM usage (**up to 30x less than Elasticsearch**)
  - Fast full-text search over large volumes of logs
  - Works perfectly with high-cardinality log fields (user_id, trace_id, ip)

VictoriaLogs

VictoriaLogs

- Open source database for logs

VictoriaLogs

- Open source database for logs
- Easy to setup and operate - a single small executable, which runs optimally with default configs

VictoriaLogs

- Open source database for logs
- Easy to setup and operate - a single small executable, which runs optimally with default configs
- Automatically scales to available CPU and RAM - from Raspberry PI to hosts with hundreds of CPU cores and terabytes of RAM

VictoriaLogs

- Open source database for logs
- Easy to setup and operate - a single small executable, which runs optimally with default configs
- Automatically scales to available CPU and RAM - from Raspberry PI to hosts with hundreds of CPU cores and terabytes of RAM
- Supports popular log shipping protocols - syslog, elasticsearch, loki, vector, filebeat, fluentbit, logstash, opentelemetry, telegraf - https://docs.victoriametrics.com/victorialogs/data-ingestion/

VictoriaLogs: querying features

- Provides simple yet powerful query language - LogsQL - https://docs.victoriametrics.com/victorialogs/logsql/

VictoriaLogs: querying features

- Provides simple yet powerful query language - LogsQL - https://docs.victoriametrics.com/victorialogs/logsql/
- Provides an interactive command-line interface for querying - https://docs.victoriametrics.com/victorialogs/querying/vlogscli/

VictoriaLogs: querying features

- Provides simple yet powerful query language - LogsQL - https://docs.victoriametrics.com/victorialogs/logsql/
- Provides an interactive command-line interface for querying - https://docs.victoriametrics.com/victorialogs/querying/vlogscli/
- Provides web UI for querying - https://docs.victoriametrics.com/victorialogs/querying/#web-ui

VictoriaLogs: querying features

- Provides simple yet powerful query language - LogsQL - https://docs.victoriametrics.com/victorialogs/logsql/
- Provides an interactive command-line interface for querying - https://docs.victoriametrics.com/victorialogs/querying/vlogscli/
- Provides web UI for querying - https://docs.victoriametrics.com/victorialogs/querying/#web-ui
- Provides rich HTTP querying API for integration with third-party tools - https://docs.victoriametrics.com/victorialogs/querying/#http-api

VictoriaLogs: querying features

- Provides simple yet powerful query language - LogsQL - https://docs.victoriametrics.com/victorialogs/logsql/
- Provides an interactive command-line interface for querying - https://docs.victoriametrics.com/victorialogs/querying/vlogscli/
- Provides web UI for querying - https://docs.victoriametrics.com/victorialogs/querying/#web-ui
- Provides rich HTTP querying API for integration with third-party tools - https://docs.victoriametrics.com/victorialogs/querying/#http-api
- Supports “**tail -f**” functionality for query results - https://docs.victoriametrics.com/victorialogs/querying/vlogscli/#live-tailing

LogsQL: VictoriaLogs query language

LogsQL: VictoriaLogs query language

- Very easy to learn and use

LogsQL: VictoriaLogs query language

- Very easy to learn and use
- Optimized for typical log analysis tasks over **structured and unstructured** logs

LogsQL: VictoriaLogs query language

- Very easy to learn and use
- Optimized for typical log analysis tasks over **structured and unstructured** logs
- Supports data extraction and transformation at query time

LogsQL: VictoriaLogs query language

- Very easy to learn and use
- Optimized for typical log analysis tasks over **structured and unstructured** logs
- Supports data extraction and transformation at query time
- Supports powerful analytics

LogsQL examples

Select all the logs

*

Select all the logs with the “error” word

```
error
```

https://docs.victoriametrics.com/victorialogs/logsql/#word-filter

Select all the logs with the “error” word over the last 5 minutes

```
_time:5m error
```

https://docs.victoriametrics.com/victorialogs/logsql/#time-filter

Select all the logs with the “error” or “warning” word over the last 5 minutes

```
_time:5m (error or warning)
```

https://docs.victoriametrics.com/victorialogs/logsql/#logical-filter

Select all the logs with the “error” word over the last 5 minutes, which do not contain “Failed to process” phrase

```
_time:5m error -"Failed to process"
```

Select all the logs with the “error” word over the last 5 minutes for containers with the name “fluentbit-gke”

```
_time:5m error kubernetes_container_name:fluentbit-gke
```

Select all the logs with the “error” word over the last 5 minutes for containers with the name “fluentbit-gke” using log stream filter (optimized version)

```
_time:5m error {kubernetes_container_name="fluentbit-gke"}
```

https://docs.victoriametrics.com/victorialogs/keyconcepts/#stream-fields

Select all the logs with IP addresses over the last 5 minutes

```
_time:5m ~"([0-9]+[.]){3}[0-9]+"
```

https://docs.victoriametrics.com/victorialogs/logsql/#regexp-filter

Count the number of logs for the last hour

```
_time:1h | count()
```

https://docs.victoriametrics.com/victorialogs/logsql/#count-stats

Select top 10 IP addresses seen in logs over the last week

```
_time:7d `remoteAddr="`
  | extract `remoteAddr="<ip>:`
  | stats by (ip) count() as rows
  | sort by (rows desc) limit 10
```

https://docs.victoriametrics.com/victorialogs/logsql/#extract-pipe
https://docs.victoriametrics.com/victorialogs/logsql/#stats-pipe
https://docs.victoriametrics.com/victorialogs/logsql/#sort-pipe

Select top 10 IP addresses seen in logs over the last week (simplified version)

```
_time:7d `remoteAddr="`
  | extract `remoteAddr="<ip>:`
  | top 10 by (ip)
```

https://docs.victoriametrics.com/victorialogs/logsql/#top-pipe

Select top 5 container names with the biggest number of logs with the “error” word over the last hour

```
_time:1h error | top 5 by (kubernetes_container_name)
```

Select top 5 container names with the biggest errors rate over the last hour

```
_time:1h
  | stats by (kubernetes_container_name)
    count() as total,
    count() if (error) as errors
  | math errors / total as error_rate
  | filter error_rate:(>0 <1)
  | sort by (error_rate desc) limit 5
```

https://docs.victoriametrics.com/victorialogs/logsql/#math-pipe
https://docs.victoriametrics.com/victorialogs/logsql/#filter-pipe

Read LogsQL docs!

https://docs.victoriametrics.com/victorialogs/logsql/

VictoriaLogs: real production case numbers

VictoriaLogs: real production case numbers

- Logs for the last year from our internal Kubernetes staging cluster
  - Rows: 1.9 billion (vl_storage_rows)

VictoriaLogs: real production case numbers

- Logs for the last year from our internal Kubernetes staging cluster
  - Rows: 1.9 billion (vl_storage_rows)
  - Disk space usage: 45GiB (vl_data_size_bytes)

VictoriaLogs: real production case numbers

- Logs for the last year from our internal Kubernetes staging cluster
  - Rows: 1.9 billion (vl_storage_rows)
  - Disk space usage: 45GiB (vl_data_size_bytes)
  - Uncompressed size of logs: 2.5TB (compression ratio: **55x**) (vl_uncompressed_data_size_bytes)

VictoriaLogs: real production case numbers

- Logs for the last year from our internal Kubernetes staging cluster
  - Rows: 1.9 billion (vl_storage_rows)
  - Disk space usage: 45GiB (vl_data_size_bytes)
  - Uncompressed size of logs: 2.5TB (compression ratio: **55x**) (vl_uncompressed_data_size_bytes)
  - RAM usage: 250MB (process_resident_memory_anon_bytes)

VictoriaLogs: real production case numbers

- Logs for the last year from our internal Kubernetes staging cluster
  - Rows: 1.9 billion (vl_storage_rows)
  - Disk space usage: 45GiB (vl_data_size_bytes)
  - Uncompressed size of logs: 2.5TB (compression ratio: **55x**) (vl_uncompressed_data_size_bytes)
  - RAM usage: 250MB (process_resident_memory_anon_bytes)
  - CPU usage: 5% of a single CPU core (process_cpu_seconds_total)

VictoriaLogs: real production case numbers

- Single-node VictoriaLogs container
    - CPU limits: 4

VictoriaLogs: real production case numbers

- Single-node VictoriaLogs container
  - CPU limits: 4
  - Memory limits: 14GiB

VictoriaLogs: real production case numbers

- Single-node VictoriaLogs container
  - CPU limits: 4
  - Memory limits: 14GiB
  - Persistent volume: 100GB standard (HDD-based Google Cloud disk)

VictoriaLogs: real production case numbers

- Single-node VictoriaLogs container
  - CPU limits: 4
  - Memory limits: 14GiB
  - Persistent volume: 100GB standard (HDD-based Google Cloud disk)
    - 75 read IOPS
    - 150 write IOPS
    - 12 MB/s read/write throughput

VictoriaLogs: real production case numbers

- Query performance
  - The last 100 logs with the ‘error’ word: **100ms**

VictoriaLogs: real production case numbers

- Query performance
    - The last 100 logs with the ‘error’ word: **100ms**
    - Count the number of logs with the ‘error’ word over the last day: **500ms (found ~2M logs with ‘error’ word)**

VictoriaLogs: real production case numbers

- Query performance
  - The last 100 logs with the ‘error’ word: **100ms**
  - Count the number of logs with the ‘error’ word over the last day: **500ms (found ~2M logs with ‘error’ word)**
  - Count the number of logs over the last 100 days: **300ms (found ~500M logs)**

VictoriaLogs: real production case numbers

- Query performance
  - The last 100 logs with the ‘error’ word: **100ms**
  - Count the number of logs with the ‘error’ word over the last day: **500ms (found ~2M logs with ‘error’ word)**
  - Count the number of logs over the last 100 days: **300ms (found ~500M logs)**
  - Top 5 apps with the highest log volume over the last 100 days: **500ms**

VictoriaLogs: real production case numbers

- Query performance
  - The last 100 logs with the ‘error’ word: **100ms**
  - Count the number of logs with the ‘error’ word over the last day: **500ms (found ~2M logs with ‘error’ word)**
  - Count the number of logs over the last 100 days: **300ms (found ~500M logs)**
  - Top 5 apps with the highest log volume over the last 100 days: **500ms**
  - Count the number of logs with “foobar” word across 1.9 billions of logs: **3s (found ~350 entries)**

Useful links

VictoriaLogs - https://docs.victoriametrics.com/victorialogs/

LogsQL - https://docs.victoriametrics.com/victorialogs/logsql/

How collect Kubernetes logs - https://github.com/VictoriaMetrics/helm-charts/blob/master/charts/victoria-logs-single/README.md

Ingest other logs - https://docs.victoriametrics.com/victorialogs/data-ingestion/

Interactive command-line tool for querying VictoriaLogs - https://docs.victoriametrics.com/victorialogs/querying/vlogscli/