Observability Deep Dive

Ansil H
Lead SRE@Cisco

About Me

SRE Team Lead @ Cisco

More than a decade of experience in different domains

My Talks

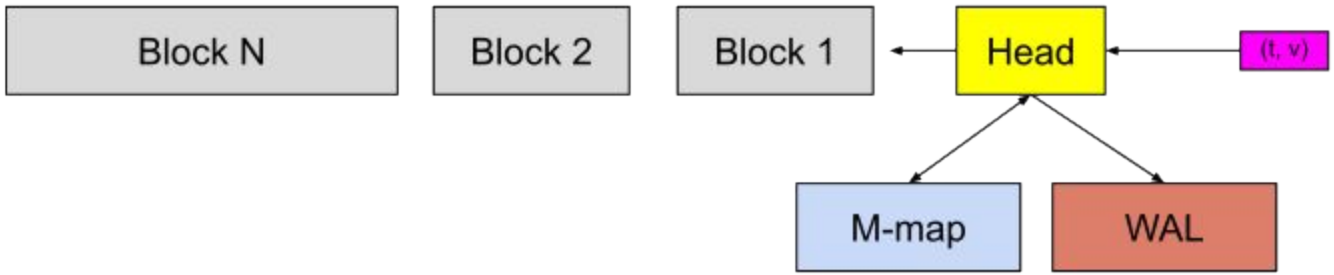
- https://github.com/ansilh/Talks

A monitoring and alerting system for distributed systems and infrastructure

Data model

TIme Series Identifier		Samples
http_requests_total	method="get" , **code="200"**	(t1,v1), (t2,v2),...
http_requests_total	method="get" , **code="400"**	(t1,v1), (t2,v2),...
Metric Name	**Labels**: Any UTF String	**Timestamp**: int64 **Values**: float64

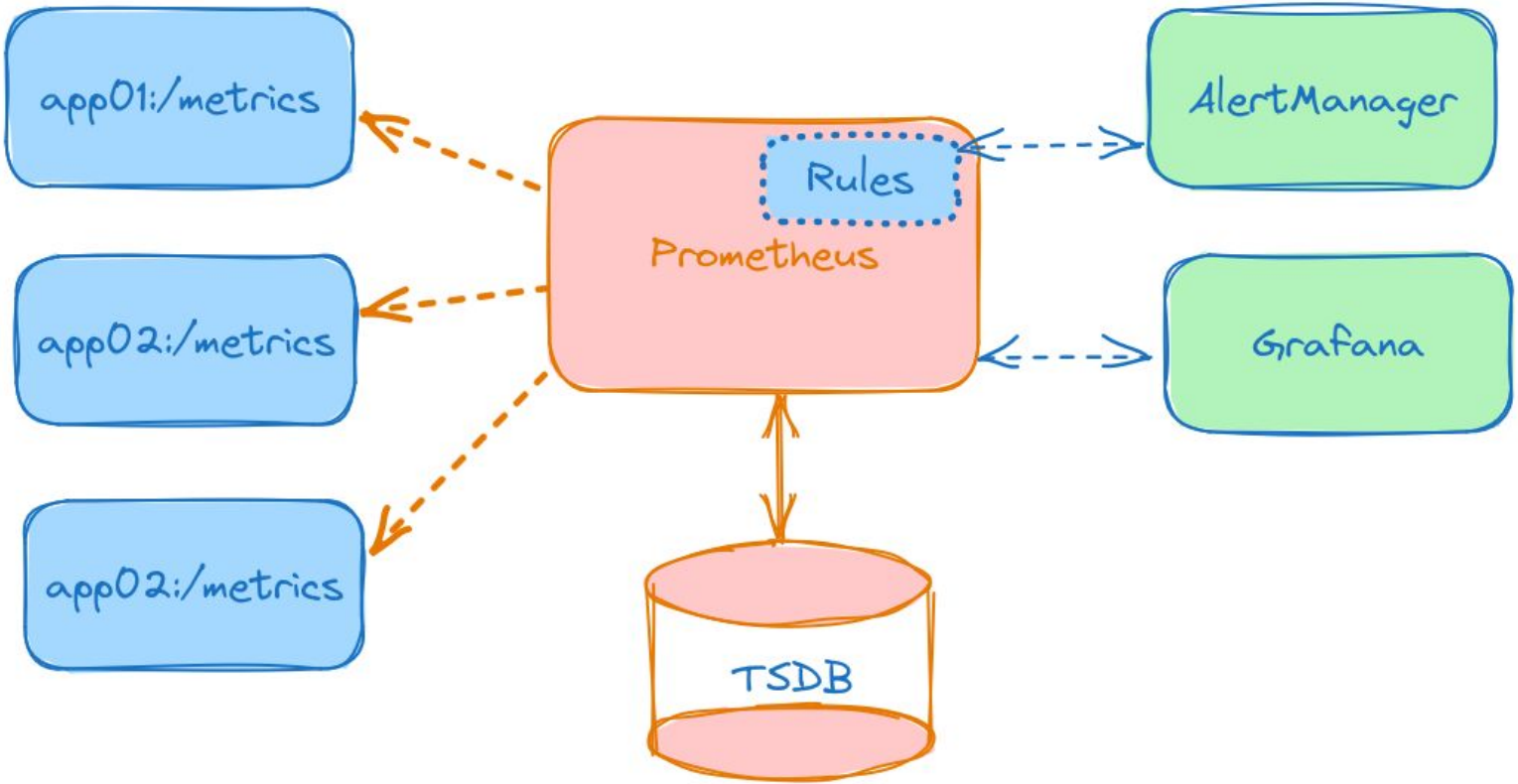
TSDB Overview



TSDB Internals

https://ganeshvernekar.com/blog/prometheus-tsdb-the-head-block/#here-is-the-entire-prometheus-tsdb-blog-series

Observability Architecture



Demo

Install Prometheus , Grafana and Alertmanager using kube-prometheus-stack

Install application

A sample web application that exposes metrics

Sample App

`https://github.com/brancz/prometheus-example-app`

- `version` - of type *gauge* - containing the app version - as a constant metric value `1` and label `version`, representing this app version
- `http_requests_total` - of type *counter* - representing the total number of incoming HTTP requests
- `http_request_duration_seconds` - of type *histogram*, representing duration of all HTTP requests
- `http_request_duration_seconds_sum` - total duration in seconds of all incoming HTTP requests
- `http_request_duration_seconds_bucket` - a histogram representation of the duration of the incoming HTTP requests

PodMonitor

PodMonitor is a CRD managed by prometheus operator that monitors Pods based on a label selector

ServiceMonitor

ServiceMonitor is a CRD managed by prometheus operator that monitors endpoints behind a service

Metric

A metric is an observable property with some defined dimensions (labels).

TimeSeries

A time series is an instance of that metric, with a unique combination of all the dimensions (labels), plus a series of timestamp & value pairs

Sample

A sample is something in between metric and time series - it’s a time series value for a specific timestamp.

Label Cardinality

How many unique values in a set.

Metric Name	Label	Value
http_requests_total	method="get" , **code="200"**	8
http_requests_total	method="get" , **code="400"**	1

This example shows a cardinality of 2

Cardinality is the number of unique combinations of all labels. The more labels you have and the more values each label can take, the more unique combinations you can create and the higher the cardinality.

Cardinality explosion

A histogram metric of 12 buckets

2 HTTP methods

7 HTTP paths

5 Instances

So that's 2x7x5x12 = **840**

If there is an additional HTTP method, HTTP path and an instance then;

3x8x6x12 = **1728**!!

Memory usage

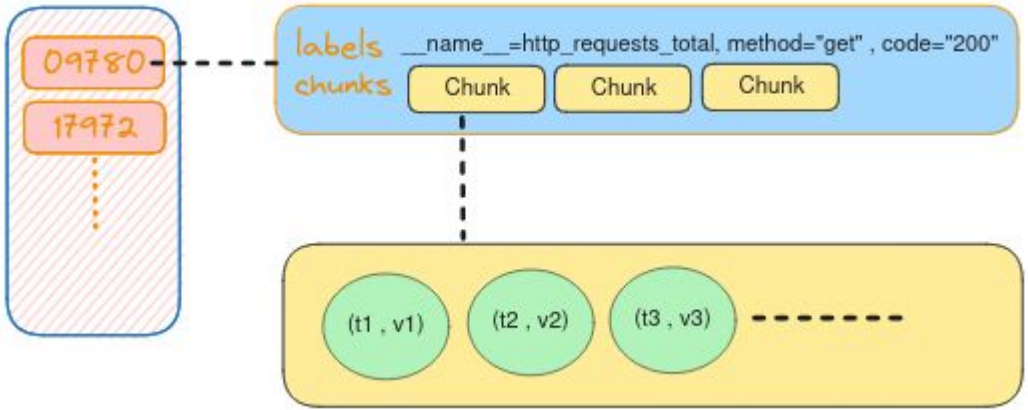
Prometheus calculates the sha256 of the collected sample and then stores the hash as a primary key inside TSDB along with the time series.

Metric Name	Label	Hash (sum hash for demo)
http_requests_total	method="get" , **code="200"**	09780
http_requests_total	method="get" , **code="400"**	17972

This helps Prometheus to quickly locate the records when storing new records.

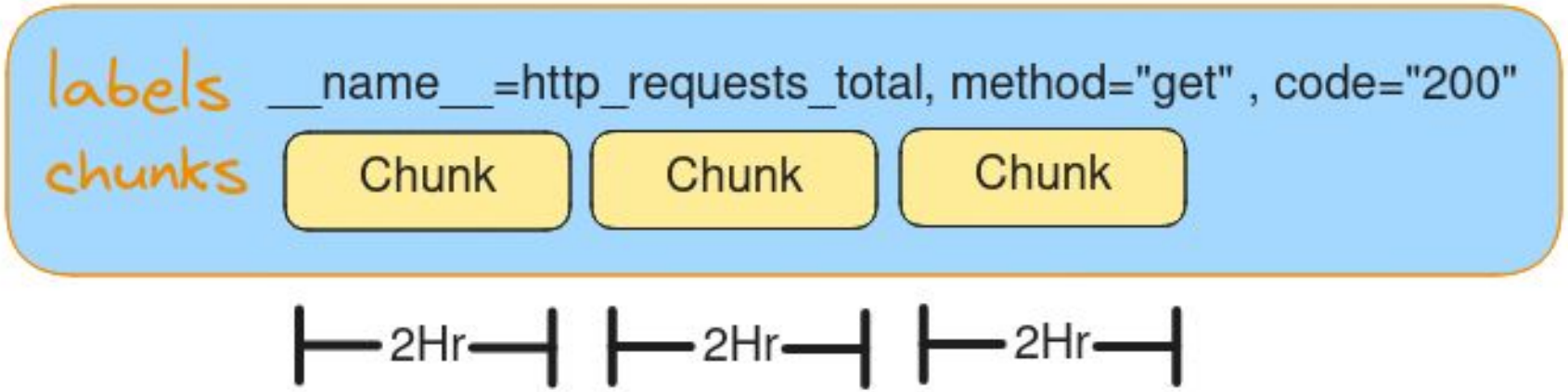
Memory Usage

The head contains the hash key and a struct called memSeries



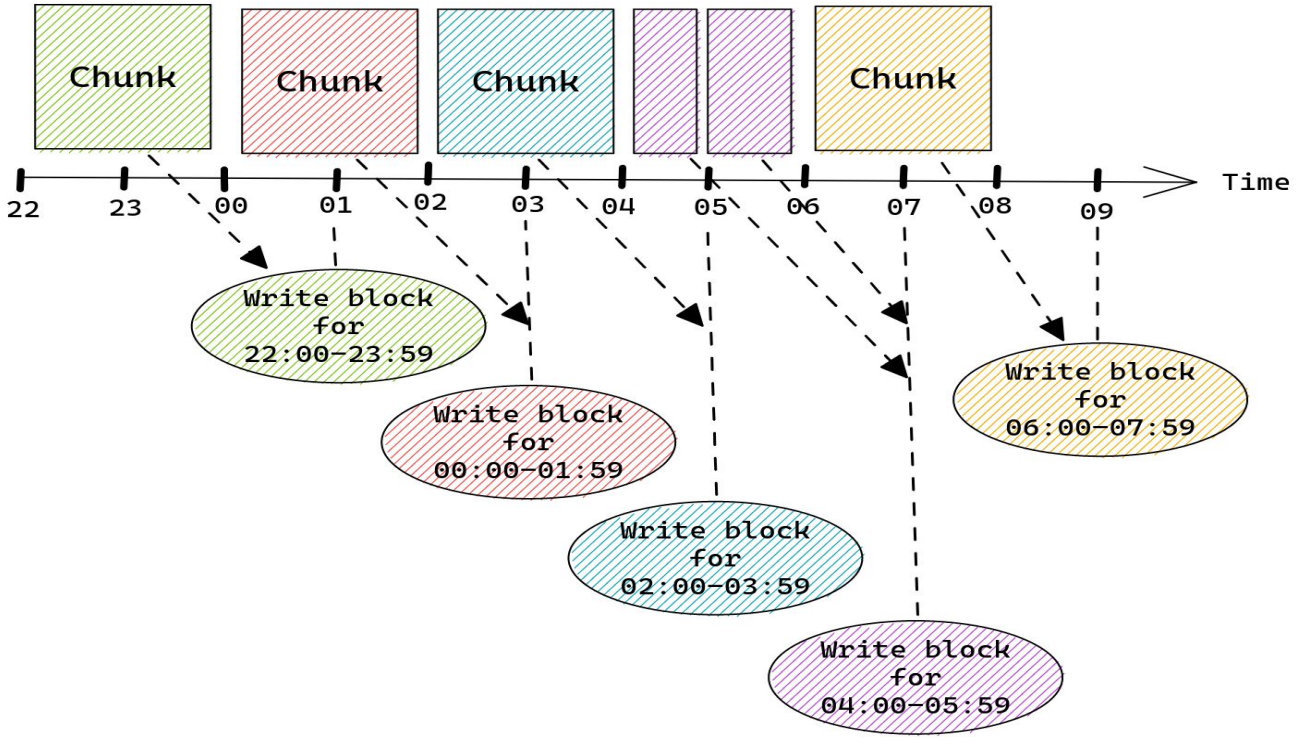
https://github.com/prometheus/prometheus/blob/v2.42.0/tsdb/head.go#L1827

memSeries


labels
__name__=http_requests_total, method="get" , code="200"
chunks
Chunk
Chunk
Chunk
2Hr
2Hr
2Hr

Memory to Disk

To reduce the memory usage, chunks will get memory mapped to disk.



Memory Usage - Conclusion

Collected samples are kept in memory

When a timeseries disappear, the labels will stay in memory (in an empty memSeris)

Older chunks are flushed to disk to reduce memory

Metrics Relabel

https://play.victoriametrics.com/select/0/target-relabel-debug

```
- regex: prometheus_replica|prometheus|endpoint
  action: labeldrop
```

```
http_requests_total{code="200", container="prometheus-example-app",endpoint="web",
instance="10.42.2.4:8080", job="monitoring/prometheus-example-app", method="get",
namespace="app", pod="prometheus-example-app-f5c769988-84dh5",
prometheus="monitoring/kube-prometheus-stack-prometheus",
prometheus_replica="prometheus-kube-prometheus-stack-prometheus-0"}
```

Step	Relabeling Rule	Input Labels	Output labels

0	`action: labeldrop`
	`regex:`
	`- prometheus_replica`
	`- prometheus`
	`- endpoint`
		http_requests_total{code="200", container="prometheus-example-app", **endpoint="web"**, instance="10.42.2.4:8080", job="monitoring/prometheus-example-app", method="get", namespace="app", pod="prometheus-example-app-f5c769988-84dh5", **prometheus="monitoring/kube-prometheus-stack-prometheus"**, **prometheus_replica="prometheus-kube-prometheus-stack-prometheus-0"**}	http_requests_total{code="200", container="prometheus-example-app", instance="10.42.2.4:8080", job="monitoring/prometheus-example-app", method="get", namespace="app", pod="prometheus-example-app-f5c769988-84dh5"}

Resulting labels: `http_requests_total{code="200", container="prometheus-example-app", instance="10.42.2.4:8080", job="monitoring/prometheus-example-app", method="get", namespace="app", pod="prometheus-example-app-f5c769988-84dh5"}`

Types of Metrics

Counter : Increase over a period of time

Eg:- Number of HTTP request etc.

Gauge : Increase of decrease over a period of time

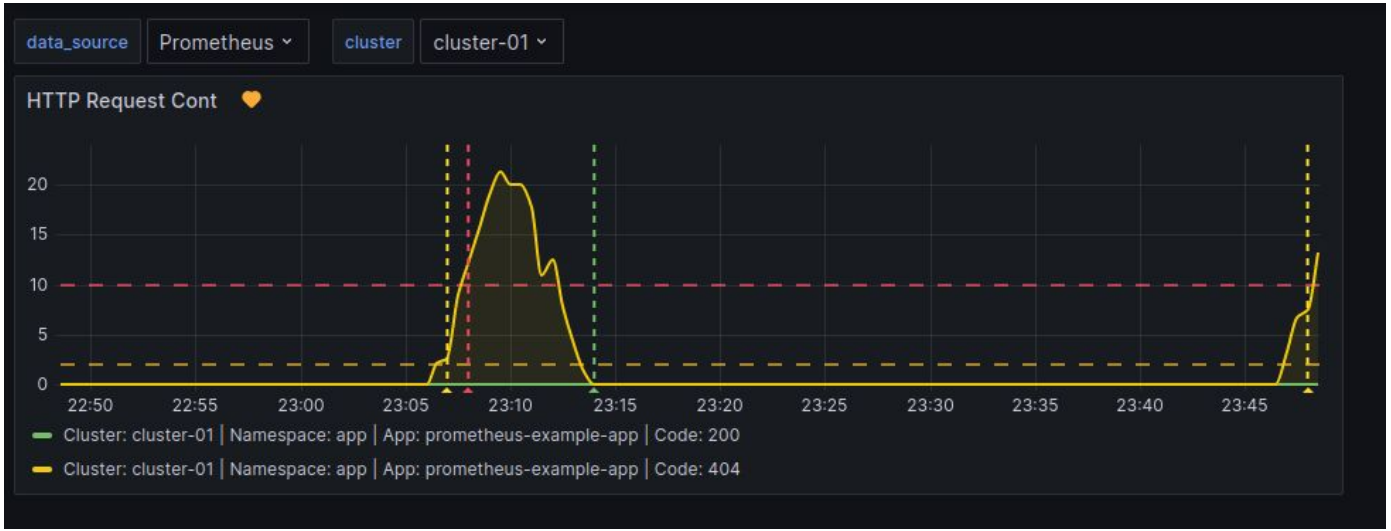
Eg:- CPU Usage, Temperature etc.

Histogram: A predefined buckets will be used for recording counter values. Each buckets will have an upper bound.

Eg:- Number of API request (100) and time taken to process each API request (0.5ms, 1ms, 2ms, 5ms, 10ms)

A tool to query, visualize, alert and explore metrics from a datasource like Prometheus

HTTP App Dashboard



Creating Dashboard

New Panel

Define Variables

Define thresholds

Queries

Variables

DataSource

Labels

Importing Dashboard to Grafana

Using Grafana ID

Using URL

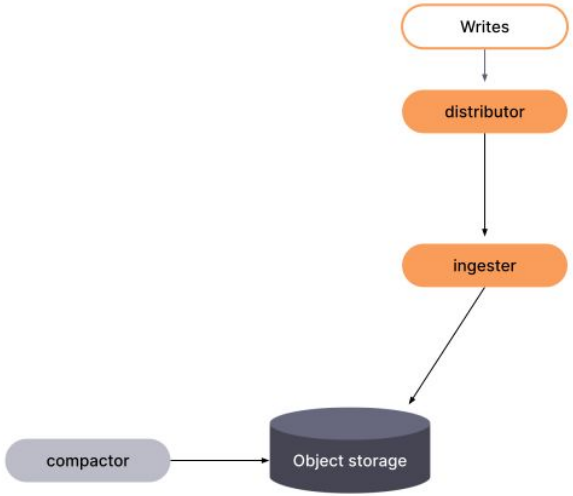
Using File

Alerting

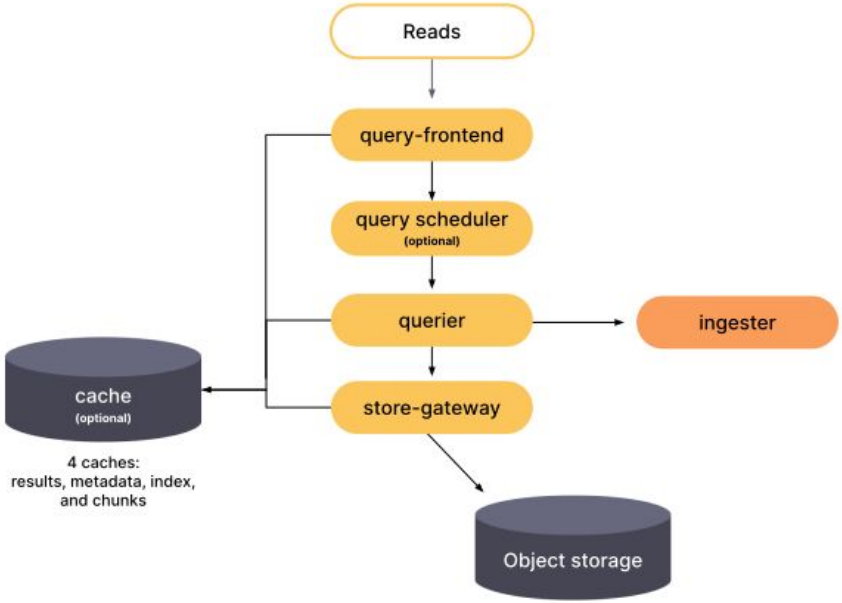
AlertManager handles alerts sent by client applications such as the Prometheus server. It takes care of deduplicating, grouping, and routing them to the correct receiver integration such as email, PagerDuty, or OpsGenie. It also takes care of silencing and inhibition of alerts.

A tool that provides a scalable long-term storage for Prometheus.

Write Path



Read Path


Reads
query-frontend
query scheduler
(optional)
querier
ingester
cache
(optional)
store-gateway
4 caches:
results, metadata, index,
and chunks
Object storage

Capacity planning

https://grafana.com/docs/mimir/latest/operators-guide/run-production-environment/planning-capacity/

Q&A

Articles related to CNCF Projects and Linux will be published soon here

https://acloudlabs.guru/post/