Prometheus/o11y workshop

Francis Begyn

September 19, 2024

o11y

Who am I

Francis Begyn

- Solutions engineer
- Open Source enthousiast and advocate
- In the monitoring space for 4 years

Intro

- Observability, what is it?
- Prometheus
  - componetns
  - configuration
- PromQL (Prometheus query language)

Observability

Observability

- Allows us to investigate problems
- Gathers insights into the inner workings of systems/programs
- Answers question *how?* and *why?*

Prometheus

Prometheus

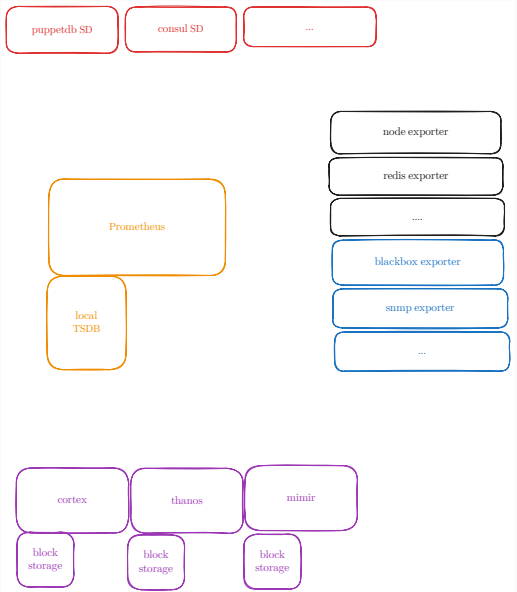
- Born in 2012 at SoundCloud
- Open Source in 2015
- Graduated CNCF project in 2018

Prometheus docs

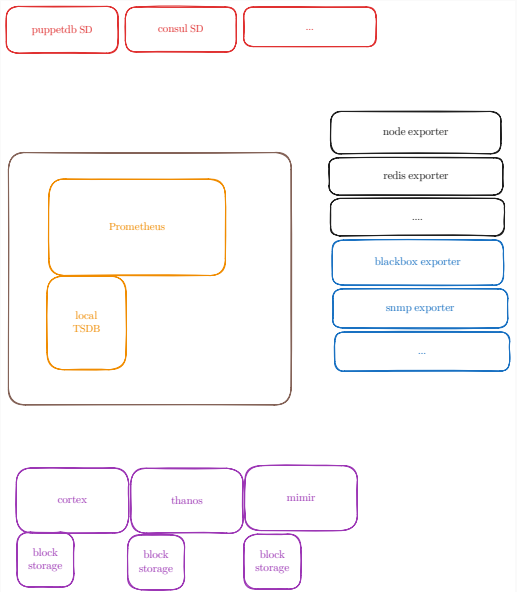
Figure 1: Prometheus docs

Prometheus components

Prometheus components



Prometheus and it's TSDB



Prometheus and it's TSDB

- Prometheus stores samples in blocks
- blocks are a collection of series over a time
  - default: min. 2h, max 6h
- blocks are persisted to the disk
- blocks once written are **immutable**

Prometheus and it's TSDB

- currently incoming samples/chunks are kept in memory
  - but still as a block!
- semi-persisted
  - Write Ahead Log (WAL)
  - WAL segments 128MB
  - min. 3 segments

Service discovery

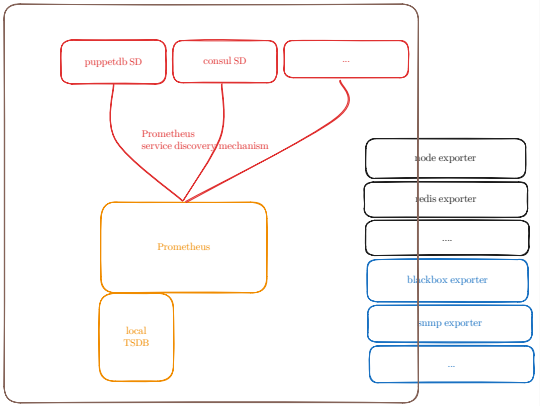
Service discovery

- created to deal with dynamic environments
  - containers
  - IaaC with cattle instead of pets

Service discovery

- created to deal with dynamic environments
  - containers
  - IaaC with cattle instead of pets
- allows for on-the-go discovery of instances of services

Prometheus targets and the service discovery



Prometheus - puppetdb

```
- job_name: node-exporter
  puppetdb_sd_configs:
  - url: http://puppet5-db.mgmtprod.inuits.eu:8080
    query: resources { type = "Package" and title = "node_exporter"
                       and environment =~ "prod|dev|testing"}
    include_parameters: true
    port: 9100
```

Output of service discovery

- the targets are fetched from the service discovery mechanism
- the service discovery mechanism returns metadata under the form of `__meta` labels

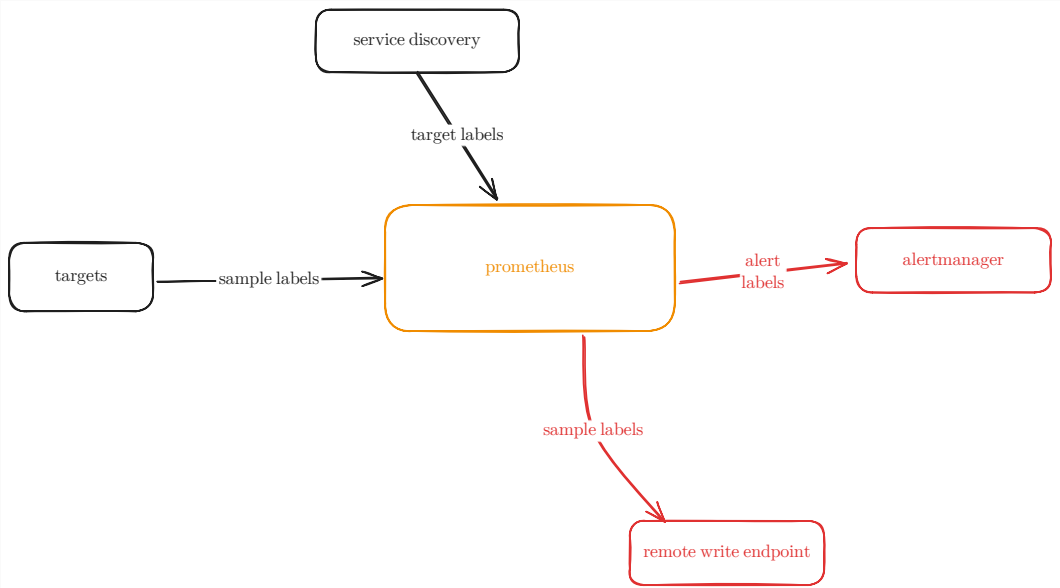
```
{
    "discoveredLabels": {
      "__address__": "foo.bar:9100", "__meta_puppetdb_certname": "foo.bar"
      "__meta_puppetdb_environment": "prod",
      ...
      "__metrics_path__": "/metrics", "__scheme__": "http",
      "__scrape_interval__": "1m", "__scrape_timeout__": "10s",
      "job": "node-exporter"
},
```

Labels … ? What are labels?

Use labels to differentiate the characteristics of the thing that is being measured

- When querying: labels are used to select which time series
- Internally prometheus: labels are used to filter and change metrics, targets, …

Labels … ? What are labels?



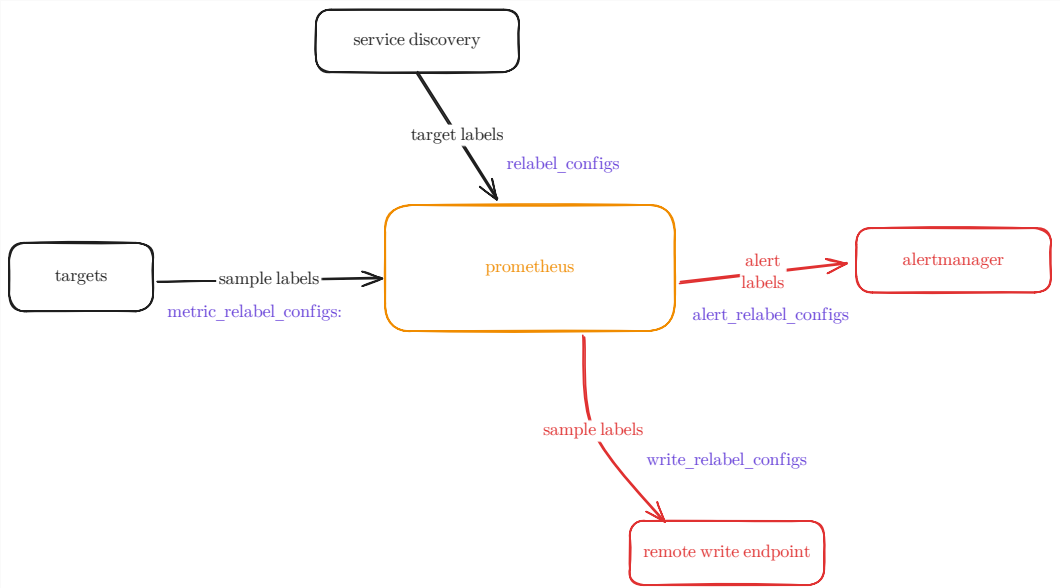
Relabeling

- relabeling consumes original set of labels and returns a new set of labels
- relabeling can also modify the reserved labels
- all labels with __ as prefix will not be visible to the end user
- https://relabeler.promlabs.com/

Relabeler webapp

Figure 6: relaber qr

Relabeling


service discovery
target labels
relabel_configs
targets
sample labels
metric_relabel_configs:
prometheus
alert
labels
alertmanager
alert_relabel_configs
sample labels
write_relabel_configs
remote write endpoint

Relabeling - an example

Relabeling rules:

```
- source_labels:
  - __meta_puppetdb_certname
  target_label: instance
- source_labels:
  - __meta_puppetdb_environment
  target_label: environment
```

Object labels:

```
"__address__": "foo.bar:9100"
"__meta_puppetdb_certname": "foo.bar"
"__meta_puppetdb_environment": "prod"
"__metrics_path__": "/metrics"
"__scheme__": "http"
"__scrape_interval__": "1m"
"__scrape_timeout__": "10s"
"job": "node-exporter"
```

Analyze Rules

Figure 8: promtheus relabel

Relabeling - an example

__address__	foo.bar:9100
__meta_puppetdb_certname	foo.bar
__meta_puppetdb_environment	prod
__metrics_path__	/metrics
__scheme__	http
__scrape_interval__	1m
__scrape_timeout__	10s
job	node-exporter

Rule 1	
action	replace
source_labels	["__meta_puppetdb_certname"]
target_label	instance

	__address__	foo.bar:9100
	__meta_puppetdb_certname	foo.bar
	__meta_puppetdb_environment	prod
	__metrics_path__	/metrics
	__scheme__	http
	__scrape_interval__	1m
	__scrape_timeout__	10s
+	instance	foo.bar
	job	node-exporter

Relabeling - an example

	__address__	foo.bar:9100
	__meta_puppetdb_certname	foo.bar
	__meta_puppetdb_environment	prod
	__metrics_path__	/metrics
	__scheme__	http
	__scrape_interval__	1m
	__scrape_timeout__	10s
+	instance	foo.bar
	job	node-exporter

↓

Rule 2

action	replace
source_labels	["__meta_puppetdb_environment"]
target_label	environment

↓

	__address__	foo.bar:9100
	__meta_puppetdb_certname	foo.bar
	__meta_puppetdb_environment	prod
	__metrics_path__	/metrics
	__scheme__	http
	__scrape_interval__	1m
	__scrape_timeout__	10s
+	environment	prod
	instance	foo.bar
	job	node-exporter

Reserved labels

- allows for dynamic changes to several scrape settings
- `__address__`: the actual endpoint prometheus will fetch from
- `__metrics_path____`: the url path appended to `__address__`
- `__scheme__`: the protocol scheme HTTP(S)
- `__scrape_interval__`: how often to scrape
- `__scrape_timeout__`: how long a scrape is allowed to take

Excersise

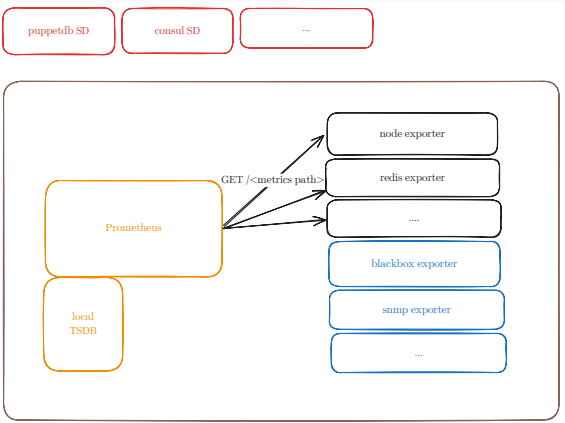
- who can tell me what's happening here?

```
- job_name: 'blackbox'
  static_configs:
    - targets:
      - https://prometheus.io   # Target to probe with https.
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 127.0.0.1:9115  # The blackbox exporter's real hostnar
```

Prometheus exporters

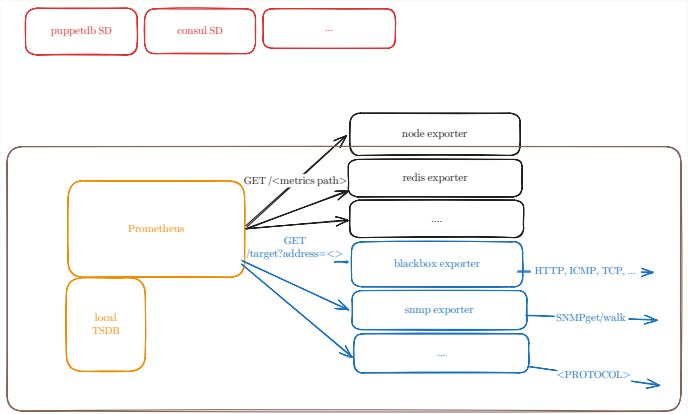
Prometheus exporters

- 2 types of exporters
  1. exporters that expose data of it's own



Prometheus exporters

- 2 types of exporters
  2. exporters that expose data they collect from other targets



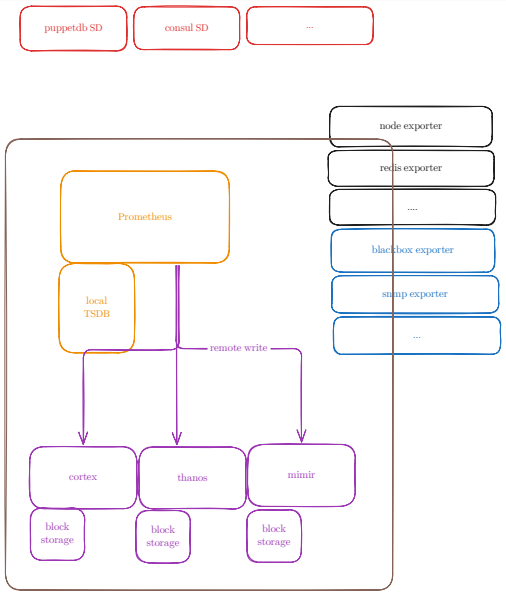
Prometheus exporters

- collect data
- expose the collected data in the Prometheus format

```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage co
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0.000138442
go_gc_duration_seconds{quantile="0.25"} 0.00022757
go_gc_duration_seconds_count 56699
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 14333
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.20.5"} 1
```

Prometheus remote write

Prometheus remote write



Prometheus remote write

- stores the prometheus blocks on object storage instead of locally
- distributed systems
  - query load
  - ingestion load
  - evaluation load

Prometheus remote write

- stores the prometheus blocks on object storage instead of locally
- distributed systems
  - query load
  - ingestion load
  - evaluation load
- all offer remote read as well
  - generally Grafana or other things
  - Prometheus has support for it as well

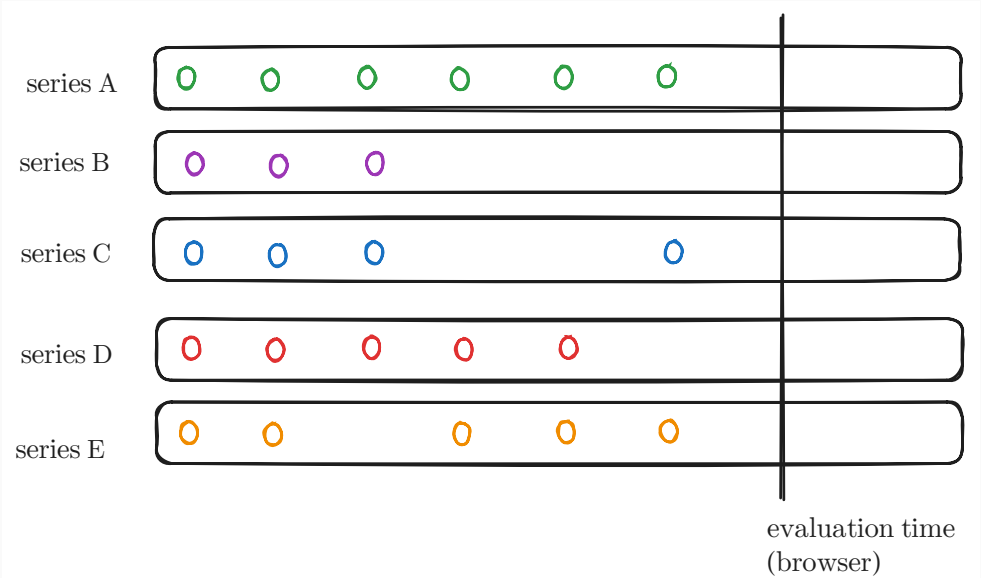
PromQL

PromQL

- prometheus query language
- function, aggregators, selectors
- `{label1=value, label2=value, foo=bar}`
- see all time series of an instance `{instance=<hostname>}`
- https://prometheus.io/docs/prometheus/latest/querying/basics/

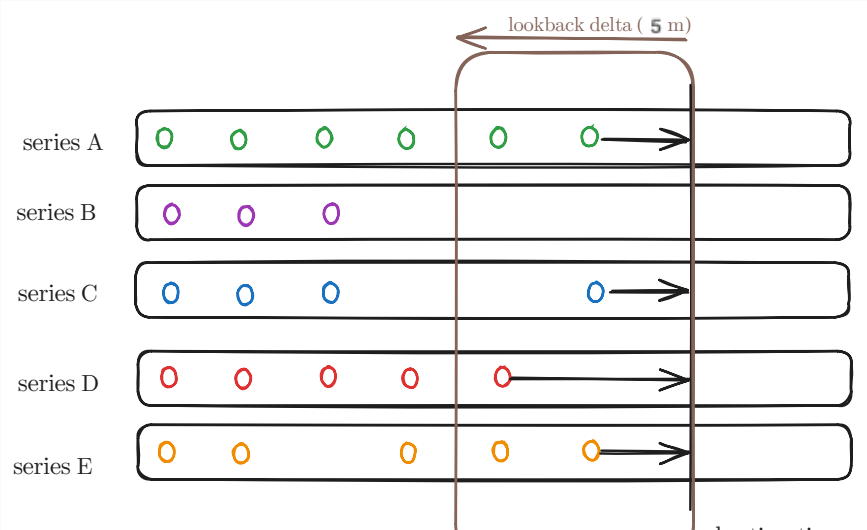
Querying

Figure 14: query docs

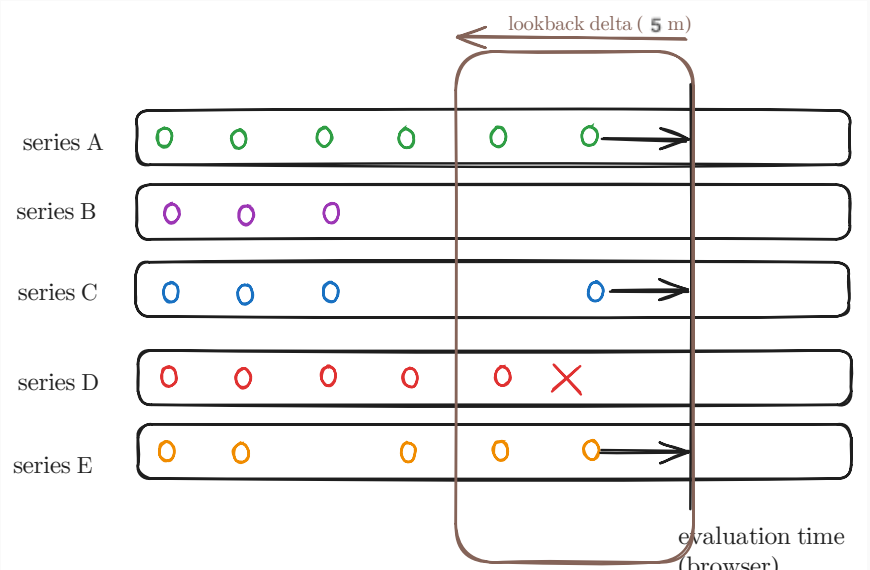

series A
series B
series C
series D
series E
evaluation time
(browser)

Instant query

instant query `node_cpu_seconds`



Stale data

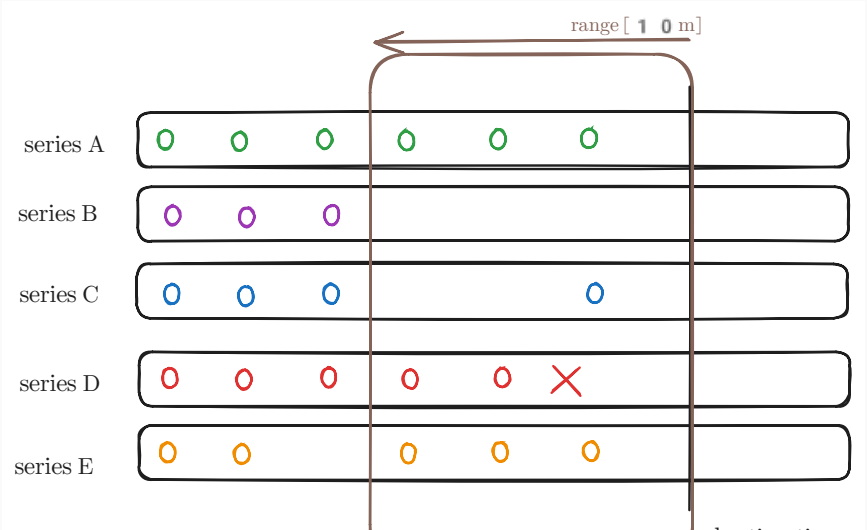

lookback delta (5 m)
series A
series B
series C
series D
series E
evaluation time
(browser)

Stale data

- series disappears from one scrape to other
- target scrape fails
- target disappears permanently
- series disappears between rule evaluations
- entire rule group disappears

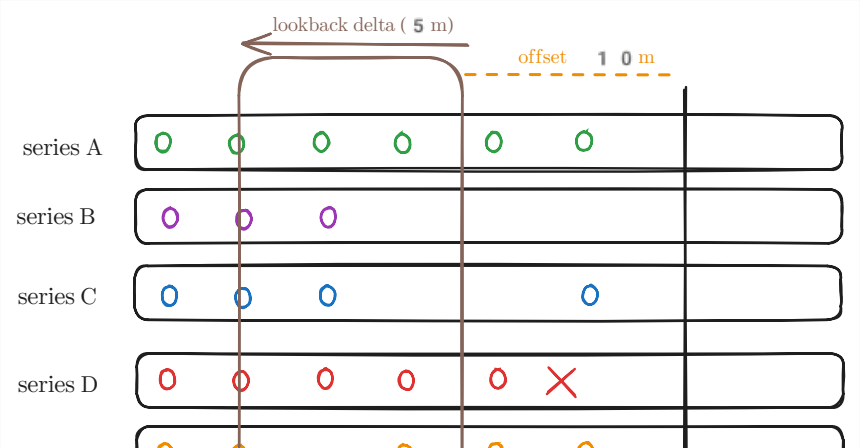
Range query

range query `node_cpu_seconds[10m]`



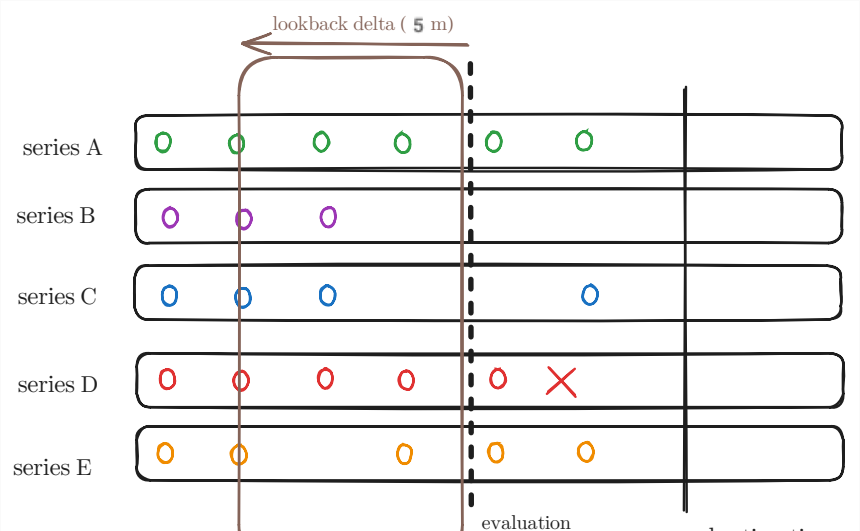
PromQL offset

- the `offset` keyword offsets the current evaluation
- + for looking back
- – for looking forward



PromQL @ modifier

Enables you to select vectors at fixed times, regardless of the current step.



Alertmanager

Alertmanager

- prometheus can generate alerts based on promql expression
- alertmanager routes these alert based on a routing tree
- “routes” can be active based on
  - time intervals
  - matching labels
  - inhibition rules

Inhibition rules

- ruleset to prevent alerts from firing based on logic
- for example:

“don’t send out the warning alert when the critical alert is already active”

Time interval

defines time intervals (to make routes active/inactive)

```
time_intervals:
- name: weekdays
  time_intervals:
  - times:
    - start_time: 08:00
      end_time: 17:00
    weekdays: ['monday:friday']
```

Time interval

```
- name: weekend
  time_intervals:
  - times:
    - start_time: 08:00
      end_time: 17:00
    weekdays: ['saturday:sunday']
- name: nightly
  time_intervals:
  - times:
    - start_time: 17:00
      end_time: 08:00
    weekdays: ['monday:sunday']
```

Receivers

Configuration blocks for webhooks, pagerduty, slack, …

```
receivers:
- name "foobar"
  email_configs:
    [ - <email_config>, ... ]
  msteams_configs:
    [ - <msteams_config>, ... ]
  pagerduty_configs:
    [ - <pagerduty_config>, ... ]
  slack_configs:
    [ - <slack_config>, ... ]
  webhook_configs:
    [ - <webhook_config>, ... ]
```

Route definition

```
[ receiver: <string> ]
# To aggregate by all possible labels use the special value '...' as the s
# group_by: ['...']
[ group_by: '[' <labelname>, ... ']' ]

# Whether an alert should continue matching subsequent sibling nodes.
[ continue: <boolean> | default = false ]
# A list of matchers that an alert has to fulfill to match the node.
matchers:
  [ - <matcher> ... ]
```

Route definition

```
[ group_wait: <duration> | default = 30s ]
[ group_interval: <duration> | default = 5m ]
[ repeat_interval: <duration> | default = 4h ]

# Times when the route should be muted.
mute_time_intervals:
  [ - <string> ...]
# Times when the route should be active.
active_time_intervals:
  [ - <string> ...]

# Zero or more child routes.
routes:
  [ - <route> ... ]
```

Any questions?

Francis Begyn

@fbegyn > Github/…

@fbegyn@social.begyn.be

https://francis.begyn.be

O11y

https://o11y.eu/prometheus-support/

Figure 21: promlabs YT