ENVIRONMENTS SETUP

BASIC REQUIREMENTS

- Network Devices (Routers/Switches/Firewalls/etc.)
  - Physical
  - Virtual
    - Virtual routers like Cisco XRv, CSR1000v
    - GNS3
    - CSR1000v from cloud providers(AWS, Azure, etc)
- Laptop (windows, Linux, Mac)
  - Install Docker https://docs.docker.com/engine/install/
  - Install docker-compose https://docs.docker.com/compose/install/

FREE OPTION

- SNMP Server
  - docker pull xiaopeng163/snmp_server:latest
- SNMP Client

进程启动

TELEGRAF系列总结

TELEGRAF的配置

TELEGRAF的官网和代码仓库

TELEGRAF的测试环境启动

正则PROCESSOR和动态标签

TAG和FIELD

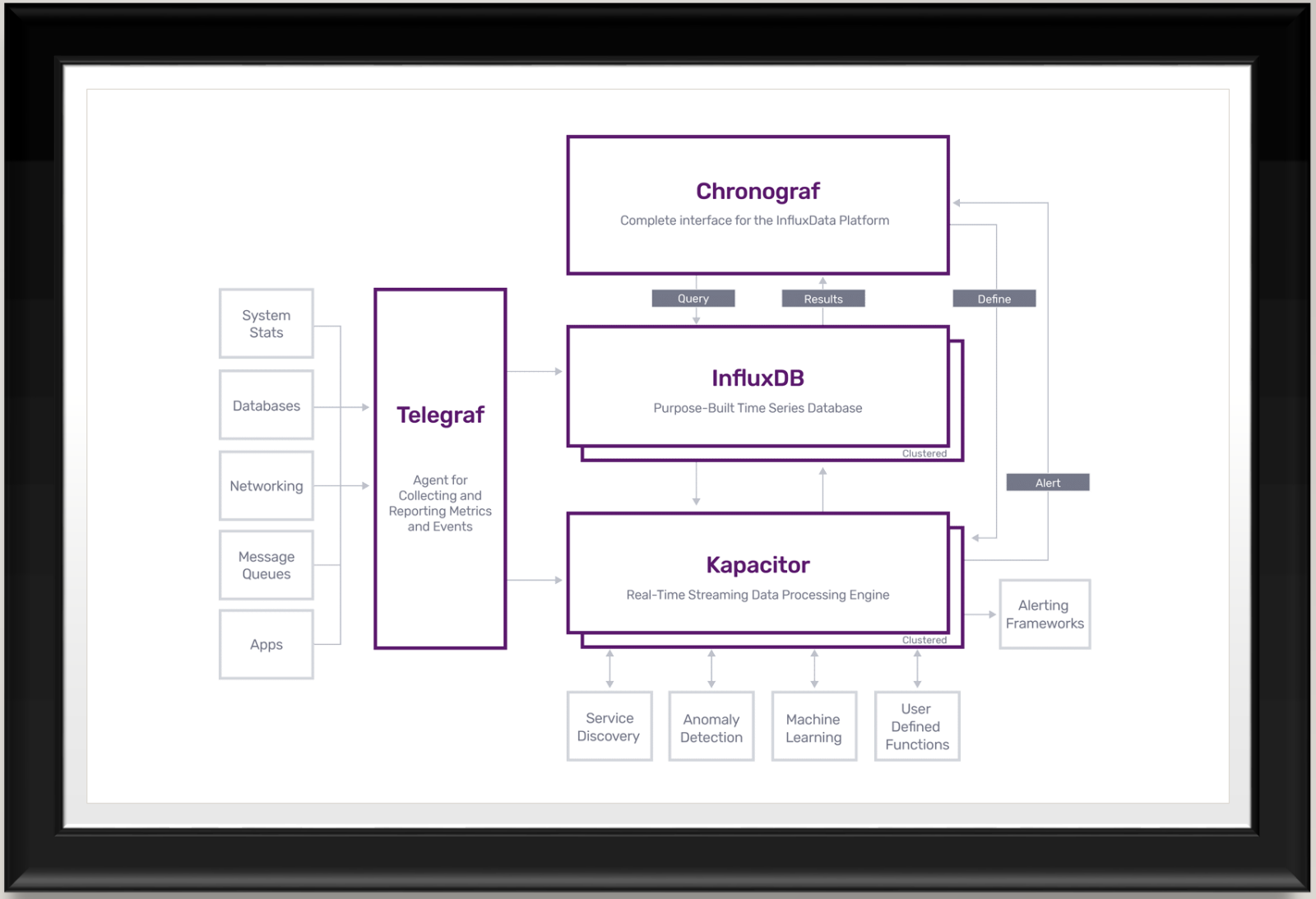
相同的INPUT插件如何配置不同的参数

通过TELEGRAF.D组织配置文件

WHAT IS TELEGRAF?

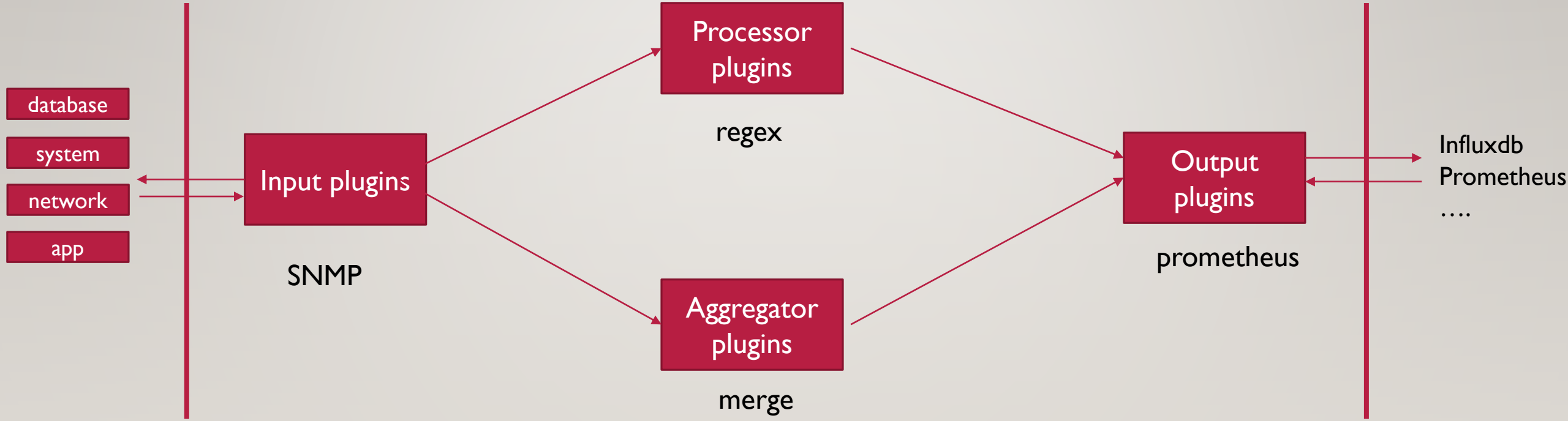
- Telegraf is a plugin-driven server agent for collecting and sending metrics and evens from databases, systems, and IoT sensors.
- Telegraf is written in Go and open sourced.

ARCHITECTURE



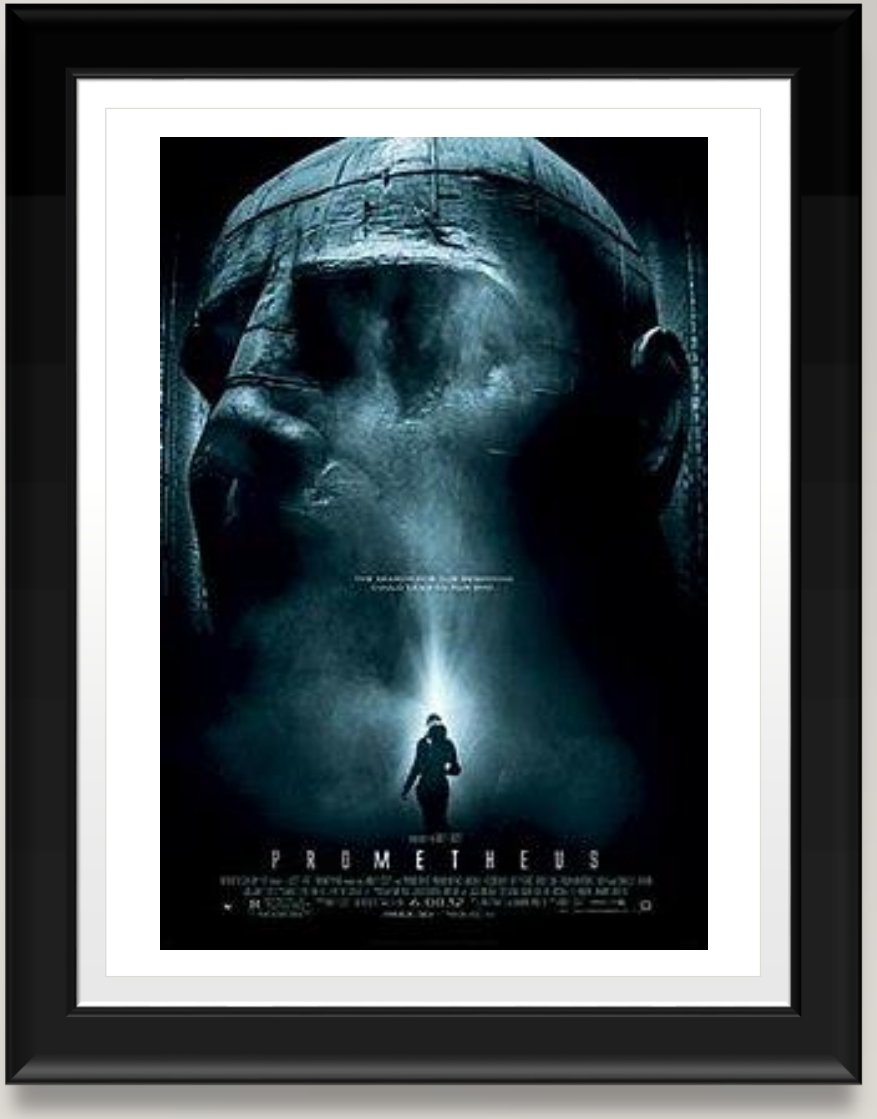
TELEGRAF PLUGINS

https://github.com/influxdata/telegraf



WHAT IS PROMETHEUS

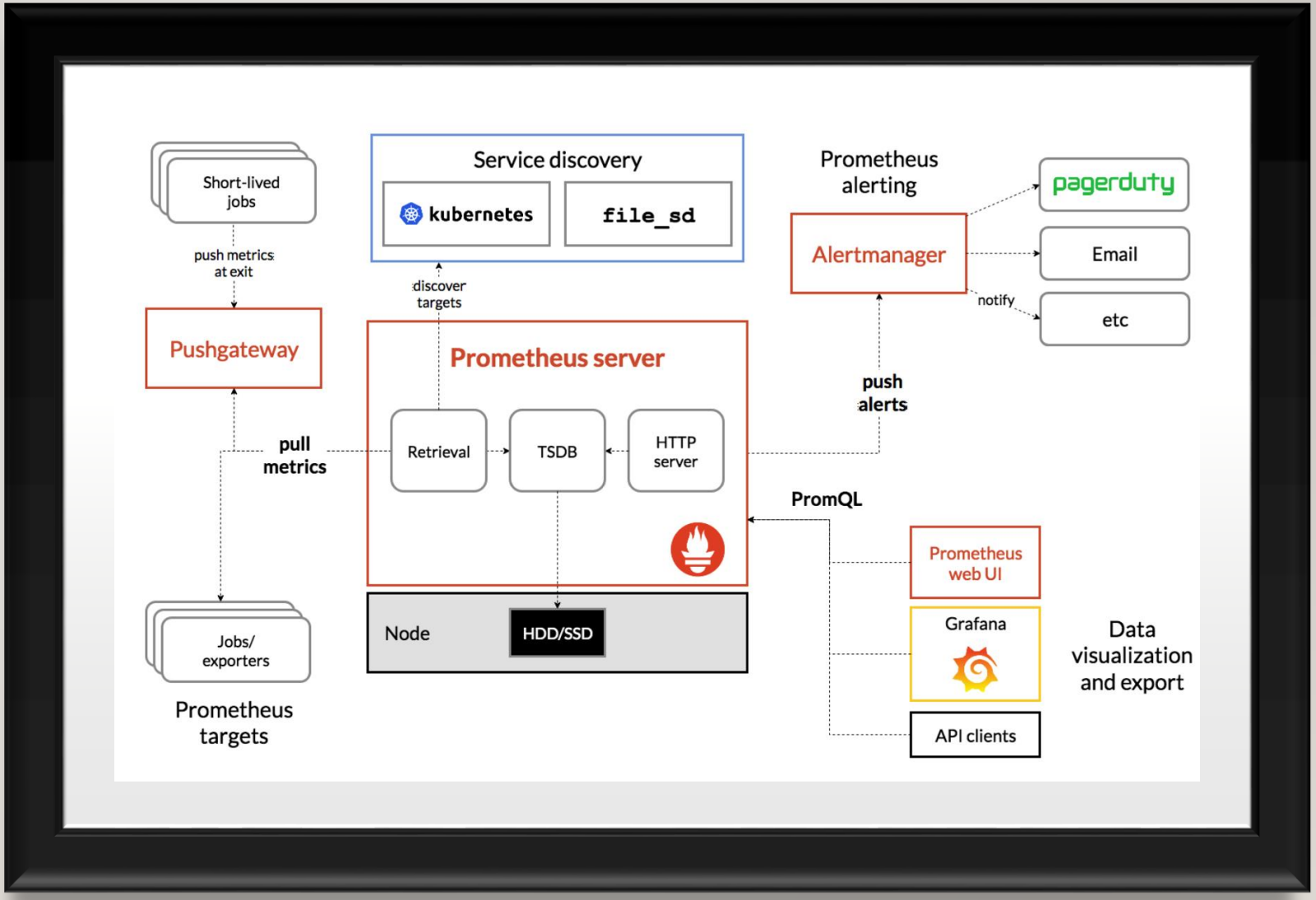
- Prometheus is an open-source systems monitoring and alerting toolkit
- Features include:
  - Time series data model
  - PromQL, a flexible query language
  - time series collection happens via a pull model over HTTP



COMPONETS

- the main Prometheus server which scrapes and stores time series data
- client libraries for instrumenting application code
- a push gateway for supporting short-lived jobs
- special-purpose exporters for services like HAProxy, StatsD, Graphite, etc.
- an alertmanager to handle alerts
- various support tools

ARCHITECTURE



PROMETHEUS QUERYING

Element	Value
ping_average_response_ms{environment="testing",host="telegraf",instance="telegraf:9273",job="telegraf",service_name="amazon",url="amazon.cn"}	215.951
ping_average_response_ms{environment="testing",host="telegraf",instance="telegraf:9273",job="telegraf",service_name="amazon",url="amazon.com"}	116.059
ping_average_response_ms{environment="testing",host="telegraf",instance="telegraf:9273",job="telegraf",service_name="amazon",url="amazon.de"}	40.057
ping_average_response_ms{environment="testing",host="telegraf",instance="telegraf:9273",job="telegraf",service_name="amazon",url="amazon.jp"}	184.513

- Metric name: ping_average_response_ms
- Labels: service_name="amazon"
- Value: 215.951
- Timestamp: see graph

METRIC TYPES

- Counters: A cumulative, monotonic metric. Counters allow the value to either go up, stay the same or be reset to 0
- Gauges: A non-monotonic metric. Gauges can go either up or down, giving the current value at any given point in time.
- Histogram: This creates multiple series for each metric name. Sampled values are put into buckets. Sum & count metrics are also generated for each sample.
- Summary: The summary is similar to histogram in that it takes samples and creates multiple metrics, including sum & count.

HISTOGRAM METRIC

```
# HELP prometheus_tsdb_compaction_chunk_range Final time range of chunks on their
# TYPE prometheus_tsdb_compaction_chunk_range histogram
prometheus_tsdb_compaction_chunk_range_bucket{le="100"} 0
prometheus_tsdb_compaction_chunk_range_bucket{le="400"} 0
prometheus_tsdb_compaction_chunk_range_bucket{le="1600"} 0
prometheus_tsdb_compaction_chunk_range_bucket{le="6400"} 0
prometheus_tsdb_compaction_chunk_range_bucket{le="25600"} 0
prometheus_tsdb_compaction_chunk_range_bucket{le="102400"} 0
prometheus_tsdb_compaction_chunk_range_bucket{le="409600"} 0
prometheus_tsdb_compaction_chunk_range_bucket{le="1.6384e+06"} 260
prometheus_tsdb_compaction_chunk_range_bucket{le="6.5536e+06"} 780
prometheus_tsdb_compaction_chunk_range_bucket{le="2.62144e+07"} 780
prometheus_tsdb_compaction_chunk_range_bucket{le="+Inf"} 780
prometheus_tsdb_compaction_chunk_range_sum 1.1540798e+09
prometheus_tsdb_compaction_chunk_range_count 780
```

SUMMARY METRIC

```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 6.66e-05
go_gc_duration_seconds{quantile="0.25"} 0.0001517
go_gc_duration_seconds{quantile="0.5"} 0.0003172
go_gc_duration_seconds{quantile="0.75"} 0.0005894
go_gc_duration_seconds{quantile="1"} 0.0043635
go_gc_duration_seconds_sum 0.360684276
go_gc_duration_seconds_count 685
```

PROMQL BASIC QUERY

- Starts with a metric name. Like *ping_average_response_ms*
- Filter with labels, label filters support four operators
  - = equal
  - != not equal
  - =~ matches regex
  - !~ doesn't match regex

RANGE VECTOR & INSTANT VECTOR

- Range vector selector: http_requests_total{job="prometheus"}[5m]
- Instant vector selector: http_requests_total{job="prometheus",group="canary"}

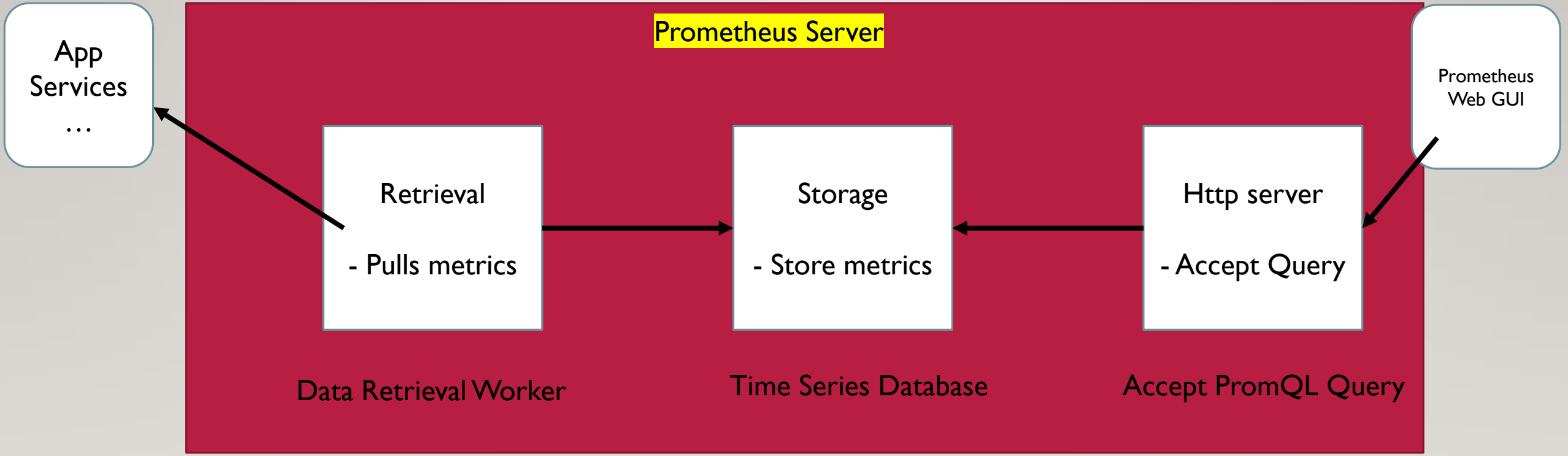
AGGREGATION OPERATORS

- https://prometheus.io/docs/prometheus/latest/querying/operators/#aggregation-operators

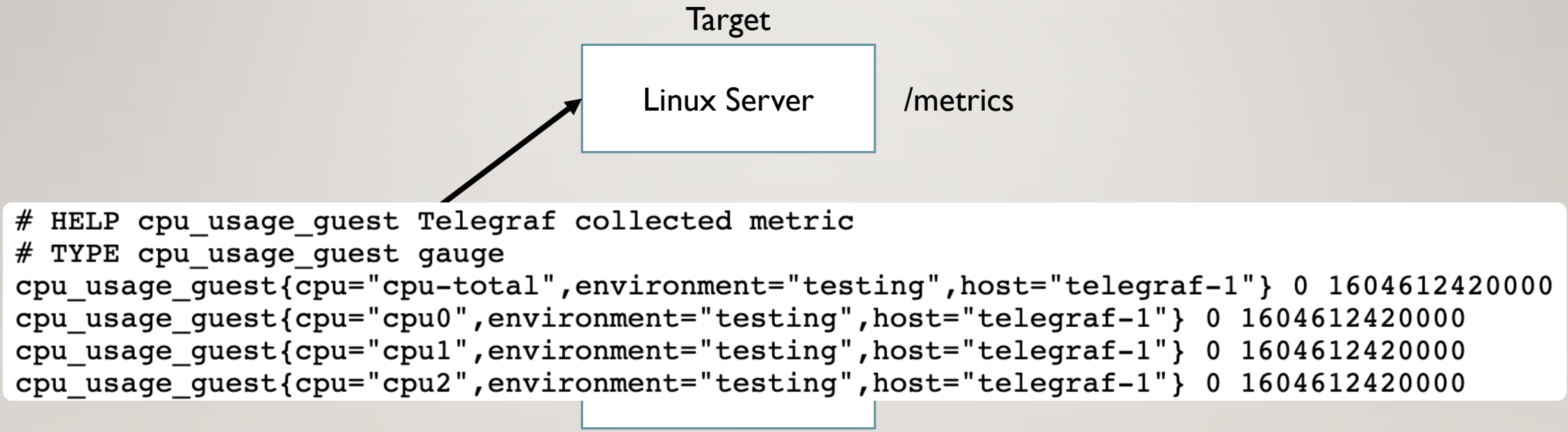
  - `sum` (calculate sum over dimensions)
  - `min` (select minimum over dimensions)
  - `max` (select maximum over dimensions)
  - `avg` (calculate the average over dimensions)
  - `group` (all values in the resulting vector are 1)
  - `stddev` (calculate population standard deviation over dimensions)
  - `stdvar` (calculate population standard variance over dimensions)
  - `count` (count number of elements in the vector)
  - `count_values` (count number of elements with the same value)
  - `bottomk` (smallest k elements by sample value)
  - `topk` (largest k elements by sample value)
  - `quantile` (calculate φ-quantile (0 ≤ φ ≤ 1) over dimensions)

告警的路由

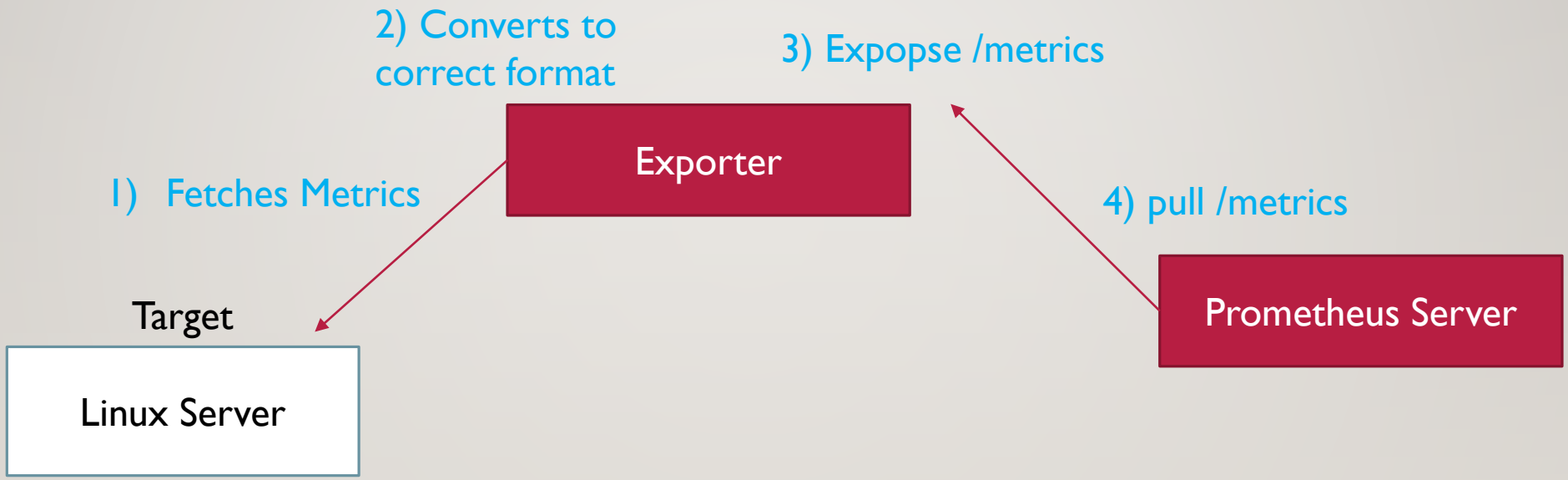
PROMETHEUS SERVER



COLLECTING METRICS



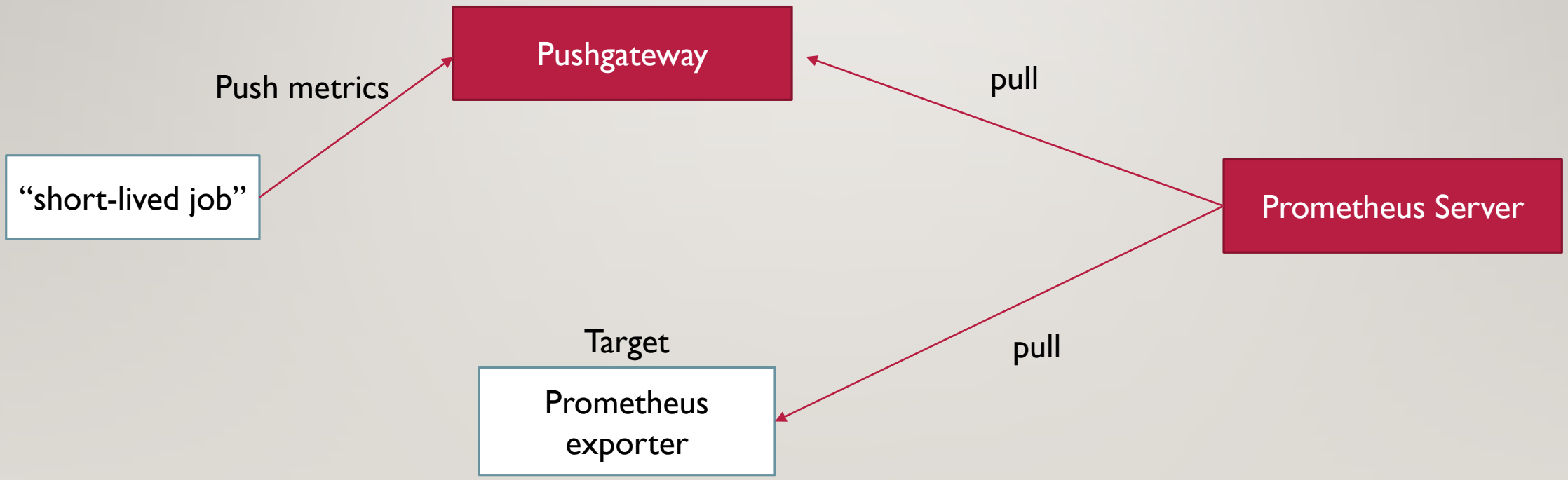
METRIC EXPORTERS



WHY PULL

- Multiple Prometheus instances can pull metrics data
- Better detection if service is up and running

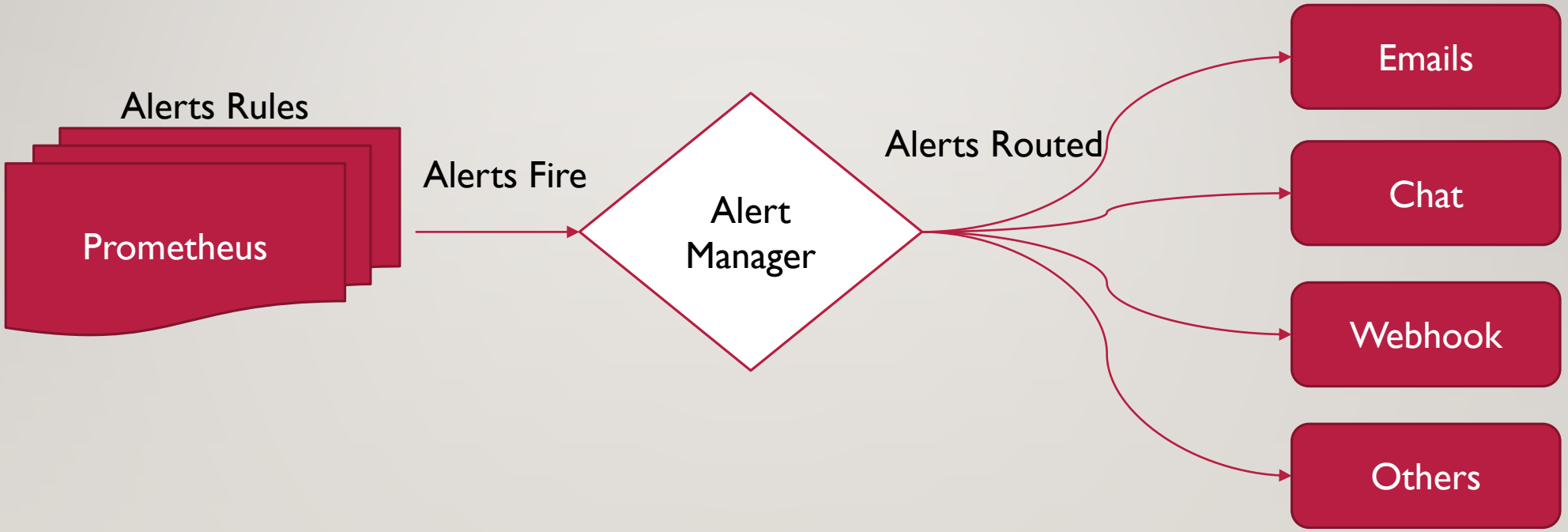
PUSHGATEWAY



ALERTING AND RULES

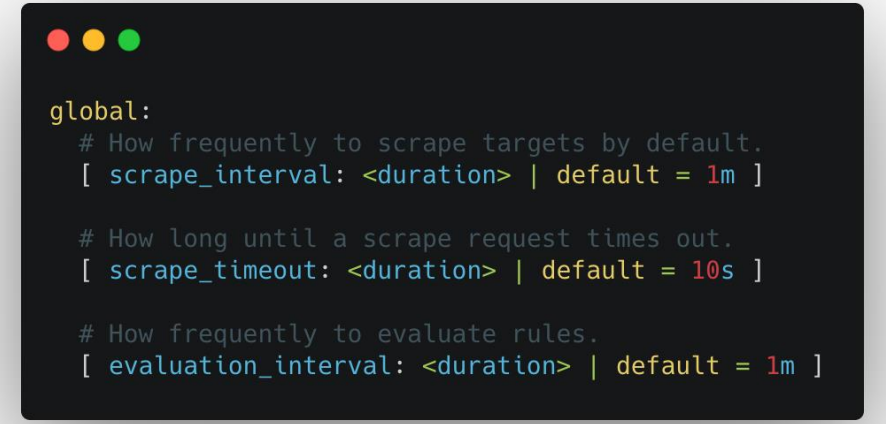
METRIC的导出和PUSH网关

ALERTING


Alerts Rules
Prometheus
Alerts Fire
Alert Manager
Alerts Routed
Emails
Chat
Webhook
Others

INTERVALS

- https://prometheus.io/docs/prometheus/latest/configuration/configuration/#configuration-file



```
global:
  # How frequently to scrape targets by default.
  [ scrape_interval: <duration> | default = 1m ]

  # How long until a scrape request times out.
  [ scrape_timeout: <duration> | default = 10s ]

  # How frequently to evaluate rules.
  [ evaluation_interval: <duration> | default = 1m ]
```

ALERT STATE

- inactive: the state of an alert that is neither firing nor pending
- pending: the state of an alert that has been active for less than the configured threshold duration (for xxxx)
- firing: the state of an alert that has been active for longer than the configured threshold duration

RECORDING RULES

GRAFANA介绍

WHAT IS GRAFANA

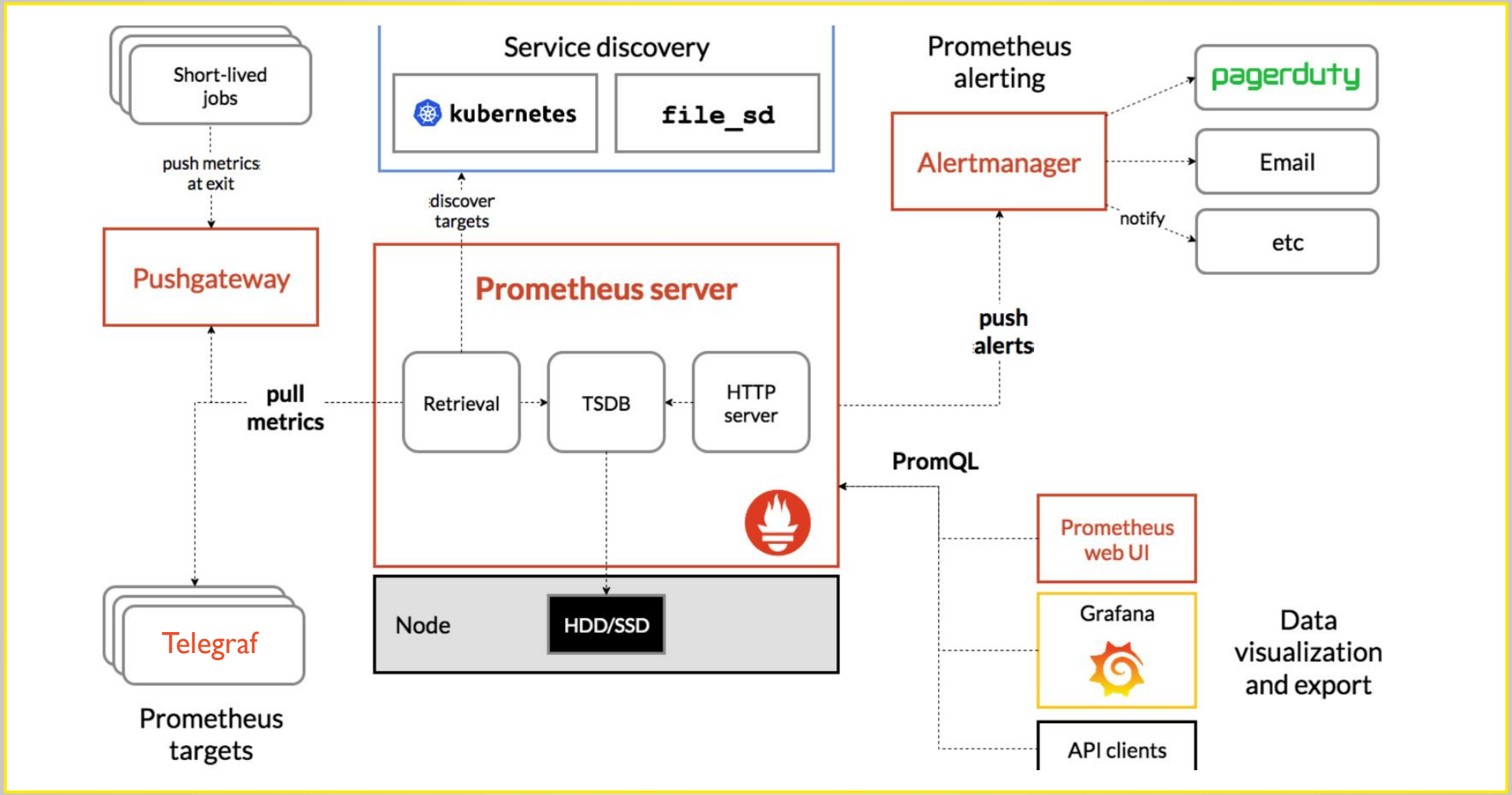
- Open source software for time series analytics(visualization, alerting)
- Supports multiple data sources like Prometheus, influxdb, etc.
- Supports graph, table, heatmap panels with many official/community-build plugins

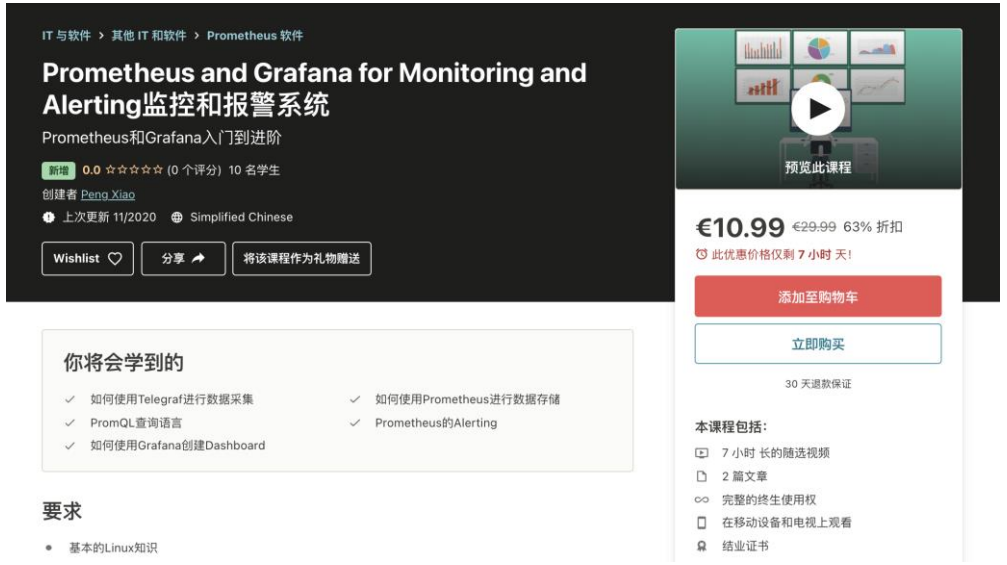
WHY USE GRAFANA

- Easy visualization
- Support 30+ data sources (Elasticsearch, InfluxDB, Prometheus, MySQL, etc.)
- Open source
- Customization
- Alert/notifications

@xiaopeng163

@xiaopeng163


Short-lived jobs
push metrics at exit
Pushgateway
Service discovery
kubernetes
file_sd
discover targets
Prometheus server
Retrieval
TSDB
HTTP server
pull metrics
Node
HDD/SSD
Telegraf
Prometheus targets
Prometheus alerting
Alertmanager
pagerduty
Email
etc
notify
push alerts
PromQL
Prometheus web UI
Grafana
API clients
Data visualization and export



Prometheus/Grafana监控与报警系统

介绍与“广告”