



云原生实战

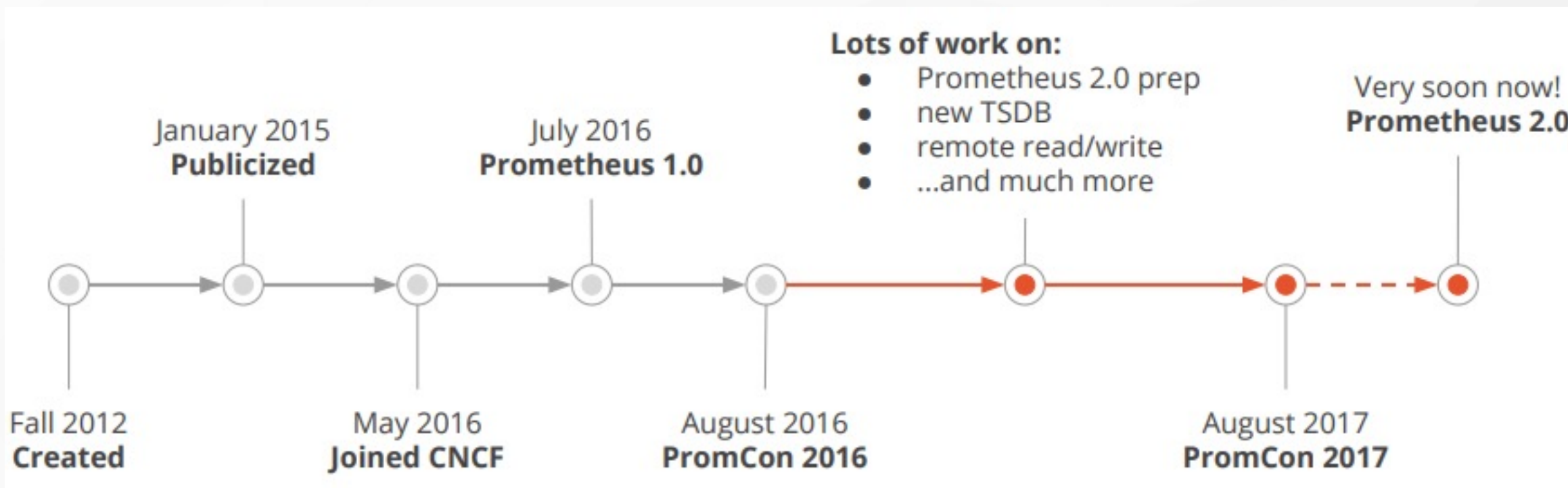
Prometheus 基础概念与上手实践

杨传胜 — KubeSphere — 开源社区经理



- 1 Prometheus 起源
- 2 Prometheus 安装使用
- 3 PromQL 介绍
- 4 Prometheus 告警处理
- 5 Prometheus Operator 安装使用与高级配置

Prometheus 起源



Prometheus 安装使用



KUBESPHERE®
_ by QingCloud



```
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

  static_configs:
    - targets: ["localhost:9090"]
```

全局配置

启动

```
$ nerdctl run -d --name prometheus --net=host \
-v /etc/prometheus:/etc/prometheus prom/prometheus
```

carbon
carbon.now.sh

规则

```
$ docker logs -f prometheus

ts=2021-11-29T02:32:18.580Z caller=main.go:406 level=info msg="No time or size retention was set so using the default time retention" duration=15d
ts=2021-11-29T02:32:18.580Z caller=main.go:444 level=info msg="Starting Prometheus" version="(version=2.31.1, branch=HEAD, revision=411021ada9ab41095923b8d2df9365b632fd40c3)"
ts=2021-11-29T02:32:18.580Z caller=main.go:449 level=info build_context="(go=go1.17.3, user=root@651e2bc2c452, date=20211105-20:19:39)"
ts=2021-11-29T02:32:18.580Z caller=main.go:450 level=info host_details="(Linux 5.13.0-21-generic #21-Ubuntu SMP Tue Oct 19 09:01:50 UTC 2021 aarch64 98e0908f5200 (none))"
ts=2021-11-29T02:32:18.580Z caller=main.go:451 level=info fd_limits="(soft=1024, hard=1024)"
ts=2021-11-29T02:32:18.580Z caller=main.go:452 level=info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2021-11-29T02:32:18.581Z caller=web.go:542 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
ts=2021-11-29T02:32:18.583Z caller=main.go:839 level=info msg="Starting TSDB ..."
ts=2021-11-29T02:32:18.585Z caller=head.go:479 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
ts=2021-11-29T02:32:18.585Z caller=head.go:513 level=info component=tsdb msg="On-disk memory mappable chunks replay completed" duration=1.708µs
ts=2021-11-29T02:32:18.585Z caller=head.go:519 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2021-11-29T02:32:18.585Z caller=head.go:590 level=info component=tsdb msg="WAL segment loaded" segment=0 maxSegment=0
ts=2021-11-29T02:32:18.585Z caller=head.go:596 level=info component=tsdb msg="WAL replay completed"
checkpoint_replay_duration=16.667µs wal_replay_duration=130.128µs total_replay_duration=170.838µs
ts=2021-11-29T02:32:18.586Z caller=main.go:866 level=info fs_type=EXT4_SUPER_MAGIC
ts=2021-11-29T02:32:18.586Z caller=main.go:869 level=info msg="TSDB started"
ts=2021-11-29T02:32:18.586Z caller=main.go:996 level=info msg="Loading configuration file" filename=/etc/prometheus/prometheus.yml
ts=2021-11-29T02:32:18.586Z caller=main.go:1033 level=info msg="Completed loading of configuration file" filename=/etc/prometheus/prometheus.yml totalDuration=446.01µs db_storage=667ns remote_storage=1µs web_handler=333ns query_engine=416ns scrape=185.754µs scrape_sd=36.418µs notify=708ns notify_sd=1.5µs rules=916ns
ts=2021-11-29T02:32:18.586Z caller=main.go:811 level=info msg="Server is ready to receive web requests."
ts=2021-11-29T02:32:18.587Z caller=web.go:195 level=info component=web msg="TLS is disabled." http2=false
```

监控目标

Prometheus 安装使用

[illegible]

Prometheus 安装使用

```
Prometheus Time Series Coll... 192.168.105.2:9090/metrics + ∞
< > ↺ ⌂ 360 直播 迅雷 钉钉 微信 192.168.105.2:9090/metrics 下载 打印 分享 收藏 更多
# 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"} 9.666e-06
go_gc_duration_seconds{quantile="0.25"} 8.5707e-05
go_gc_duration_seconds{quantile="0.5"} 0.000165737
go_gc_duration_seconds{quantile="0.75"} 0.000527495
go_gc_duration_seconds{quantile="1"} 0.004879967
go_gc_duration_seconds_sum 0.018164401
go_gc_duration_seconds_count 31

# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 35

# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.17.3"} 1

# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.48156e+07

# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 7.6942624e+07

# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.468163e+06

# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 718048

# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 1.5967816352444677e-05

# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 5.7912e+06

# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 2.48156e+07

# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
```

什么是 Exporter ?

- ❖ 从其他软件或系统中获取状态信息，并转换为 Prometheus metrics
- ❖ 软件本身通过 url 暴露 Prometheus 格式的 metrics

Prometheus 安装使用 – Node Exporter

```
$ wget https://github.com/prometheus/node_exporter/releases/download/v1.3.0/
node_exporter-1.3.0.linux-arm64.tar.gz
$ tar xzf node_exporter-1.3.0.linux-arm64.tar.gz

$ cd node_exporter-1.3.0.linux-arm64
$ ./node_exporter
```

carbon
carbon.now.sh

```
# HELP node_cpu_seconds_total Seconds the CPUs spent in each mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 109987.37
node_cpu_seconds_total{cpu="0",mode="iowait"} 3.45
node_cpu_seconds_total{cpu="0",mode="irq"} 0
node_cpu_seconds_total{cpu="0",mode="nice"} 2.42
node_cpu_seconds_total{cpu="0",mode="softirq"} 4.92
node_cpu_seconds_total{cpu="0",mode="steal"} 0
node_cpu_seconds_total{cpu="0",mode="system"} 167.97
node_cpu_seconds_total{cpu="0",mode="user"} 124.9
node_cpu_seconds_total{cpu="1",mode="idle"} 109946.29
node_cpu_seconds_total{cpu="1",mode="iowait"} 3.26
node_cpu_seconds_total{cpu="1",mode="irq"} 0
node_cpu_seconds_total{cpu="1",mode="nice"} 2.05
node_cpu_seconds_total{cpu="1",mode="softirq"} 6.22
node_cpu_seconds_total{cpu="1",mode="steal"} 0
node_cpu_seconds_total{cpu="1",mode="system"} 167.12
node_cpu_seconds_total{cpu="1",mode="user"} 124.64

# HELP node_disk_discard_time_seconds_total This is the total number of seconds spent by all discards.
# TYPE node_disk_discard_time_seconds_total counter
node_disk_discard_time_seconds_total{device="vda"} 0.152
node_disk_discard_time_seconds_total{device="vdb"} 0

# HELP node_disk_discarded_sectors_total The total number of sectors discarded successfully.
# TYPE node_disk_discarded_sectors_total counter
node_disk_discarded_sectors_total{device="vda"} 4.8255656e+07
node_disk_discarded_sectors_total{device="vdb"} 0

# HELP node_disk_discards_completed_total The total number of discards completed successfully.
# TYPE node_disk_discards_completed_total counter
node_disk_discards_completed_total{device="vda"} 3247
node_disk_discards_completed_total{device="vdb"} 0

# HELP node_disk_discards_merged_total The total number of discards merged.
# TYPE node_disk_discards_merged_total counter
node_disk_discards_merged_total{device="vda"} 0
node_disk_discards_merged_total{device="vdb"} 0

# HELP node_disk_flush_requests_time_seconds_total This is the total number of seconds spent by all flush requests.
# TYPE node_disk_flush_requests_time_seconds_total counter
node_disk_flush_requests_time_seconds_total{device="vda"} 2.636
node_disk_flush_requests_time_seconds_total{device="vdb"} 0

# HELP node_disk_flush_requests_total The total number of flush requests completed successfully
```

抓取 Node Exporter 数据

```
scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  # 采集node exporter监控数据
  - job_name: "node"
    static_configs:
      - targets: ["localhost:9100"]
```

carbon
carbon.now.sh

Prometheus 安装使用 – Node Exporter

Prometheus

Alerts

Graph

Status ▾

Help

Classic UI

Targets

All

Unhealthy

Collapse All

node (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node"	3.808s ago	10.913ms	

prometheus (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	5.350s ago	8.018ms	

Prometheus

Alerts

Graph

Status ▾

Help

Classic UI

☐ Use local time

☐ Enable query history

☒ Enable autocomplete

☒ Enable highlighting

☒ Enable linter

Q

node_load5

⌵

Execute

Table

Graph

<

Evaluation time

>

node_load5(instance="localhost:9100", job="node")

0.01

Remove Panel

Q

node_filesystem_files

⌵

Execute

Table

Graph

<

Evaluation time

>

node_filesystem_files(device="/dev/vda1", fstype="ext4", instance="localhost:9100", job="node", mountpoint="/")

3889920

node_filesystem_files(device="/dev/vda15", fstype="vfat", instance="localhost:9100", job="node", mountpoint="/boot/efi")

0

node_filesystem_files(device="/Users/carson", fstype="fuse.sshfs", instance="localhost:9100", job="node", mountpoint="/Users/carson")

6728029241

node_filesystem_files(device="/tmp/lima", fstype="fuse.sshfs", instance="localhost:9100", job="node", mountpoint="/tmp/lima")

6728029241

node_filesystem_files(device="/tmpfs", fstype="tmpfs", instance="localhost:9100", job="node", mountpoint="/run")

500517

node_filesystem_files(device="/tmpfs", fstype="tmpfs", instance="localhost:9100", job="node", mountpoint="/run/lock")

500517

node_filesystem_files(device="/tmpfs", fstype="tmpfs", instance="localhost:9100", job="node", mountpoint="/run/snappd/ns")

500517

node_filesystem_files(device="/tmpfs", fstype="tmpfs", instance="localhost:9100", job="node", mountpoint="/run/user/501")

100103

Remove Panel

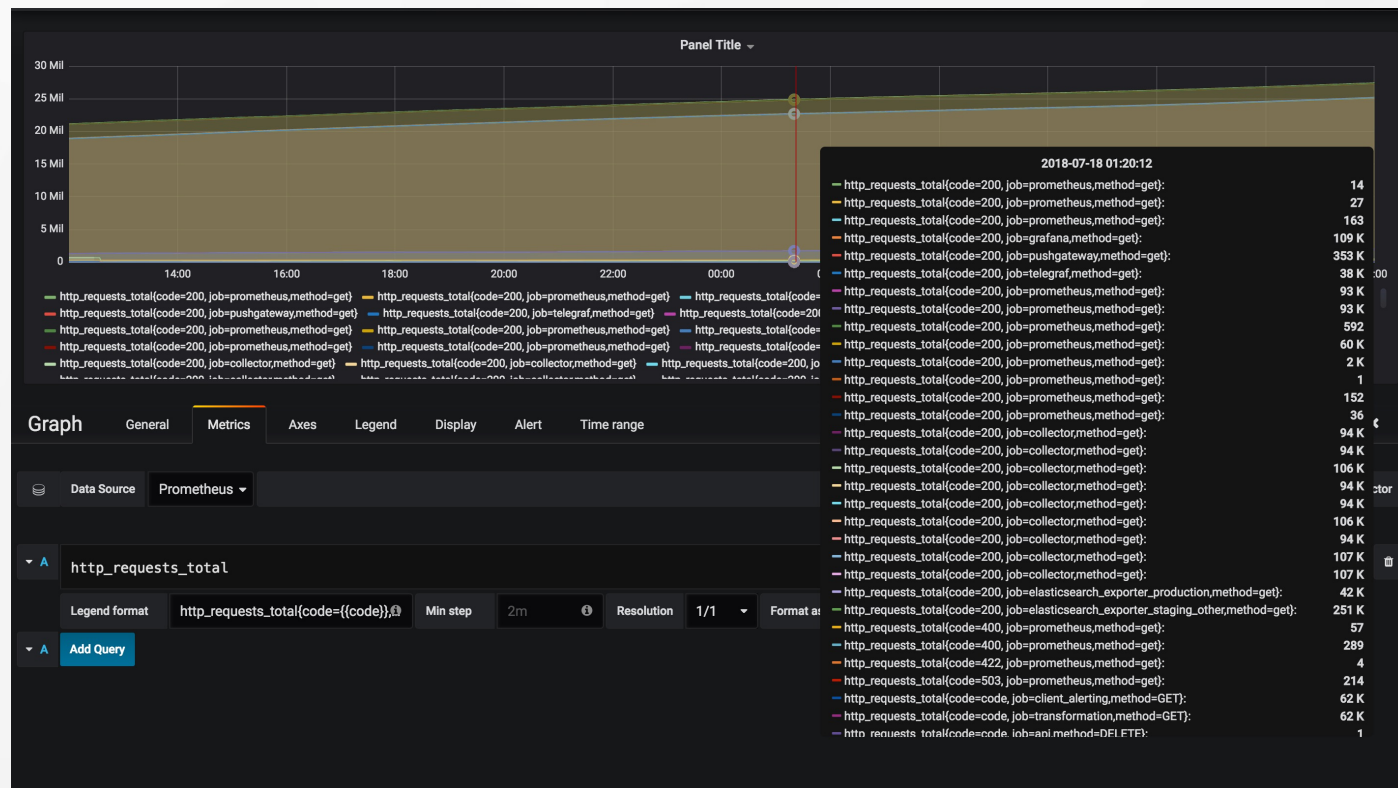
Add Panel

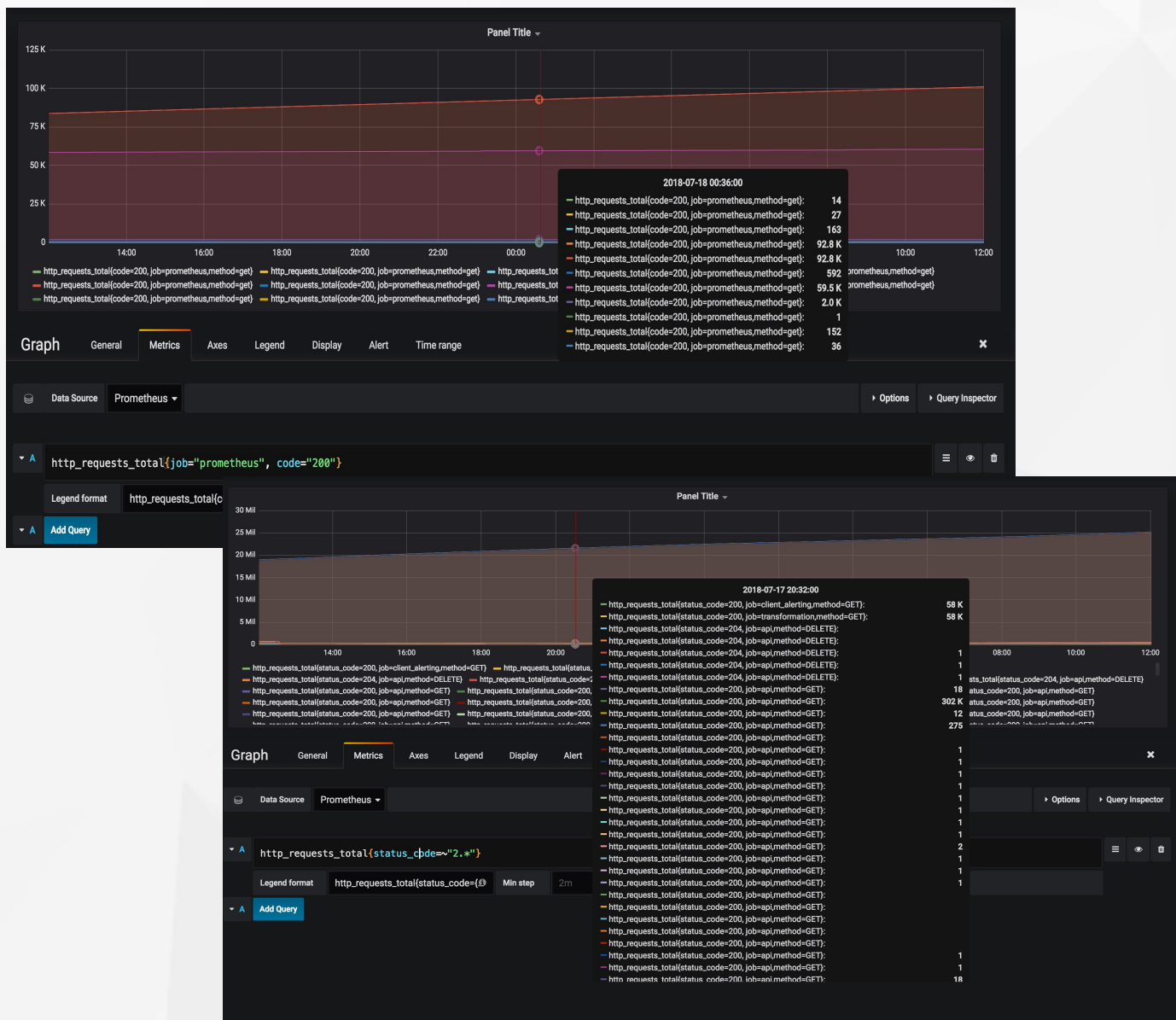
- 1. 瞬时向量 (Instant vector)**
一组时间序列，每个时间序列包含单个样本，它们共享相同的时间戳。也就是说，表达式的返回值中只会包含该时间序列中的最新的一个样本值。
- 2. 区间向量 (Range vector)**
一组时间序列，每个时间序列包含一段时间范围内的样本数据。
- 3. 标量 (Scalar)**
浮点型的数据值。
- 4. 字符串 (String)**

http_requests_total

{__name__="http_requests_total"}

- 包含所有时间序列
- 只增不减，Counter



 KUBESPHERE®
_ by QingCloud 尚硅谷

```
http_requests_total{job="prometheus"}[5m]
```

- [] 定义时间选择的范围
- 单位：s, m, h, d, w, y

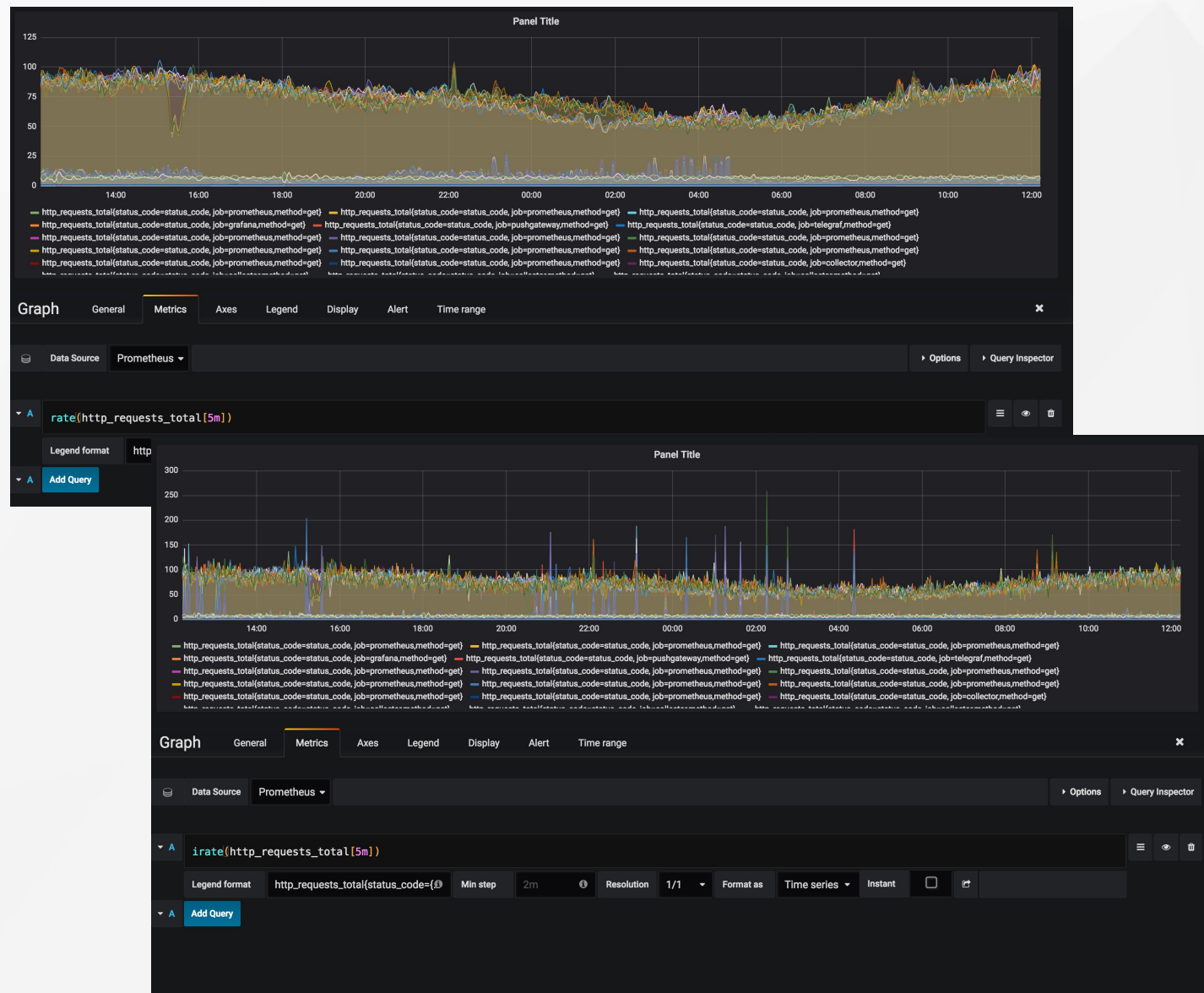
PromQL 介绍 – 区间向量

➤ `rate(http_requests_total[5m])`

区间范围内的平均增长速率

➤ `irate(http_requests_total[5m])`

通过区间向量中最后两个两本数据来计算区间向量的增长速率



➤ Counter (计数器) : 只增不减

服务的请求数、已完成的任务数、错误发生的次数

➤ Gauge (仪表盘) : 可随意变化

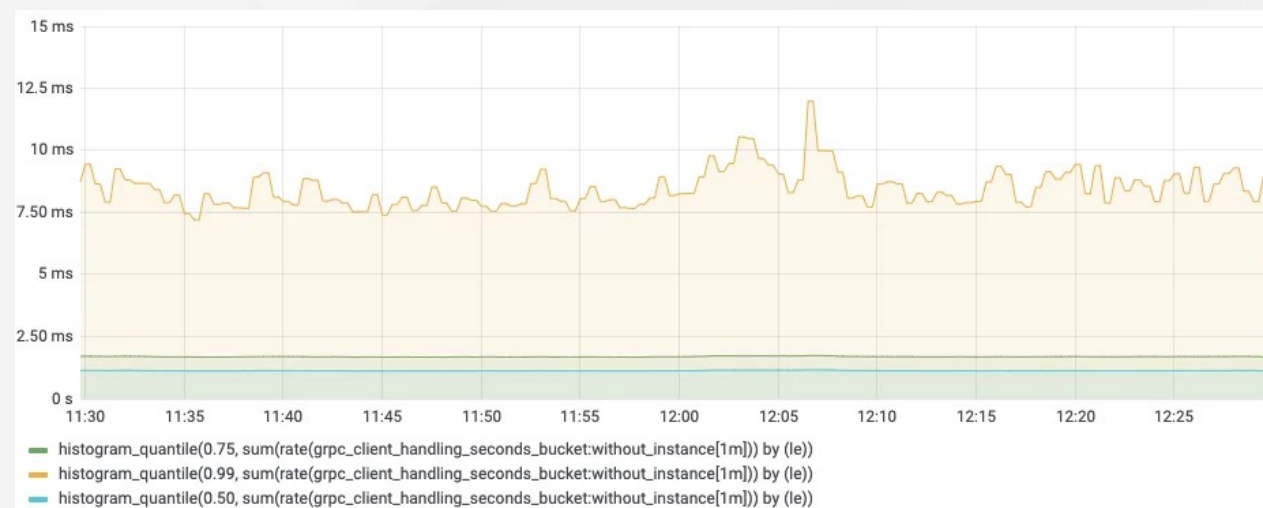
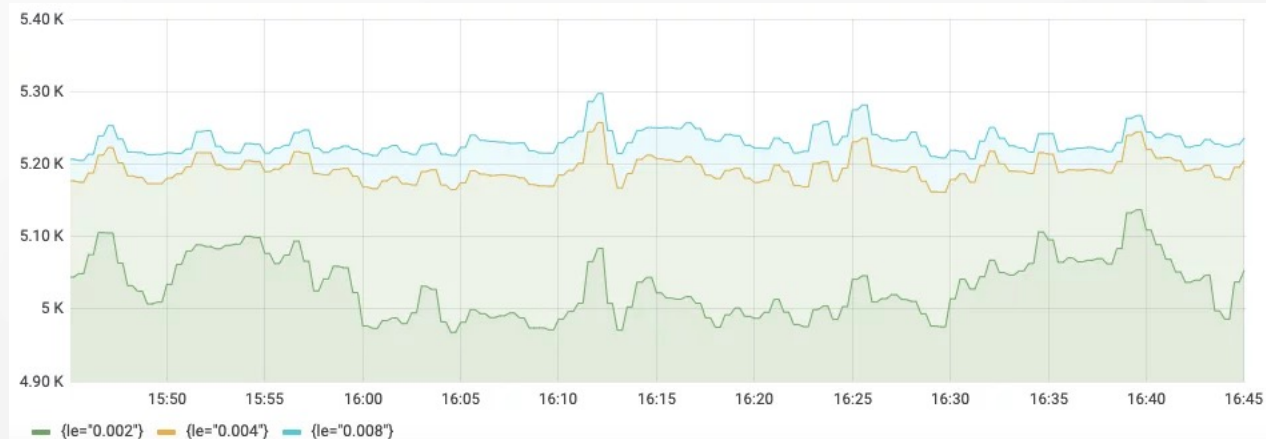
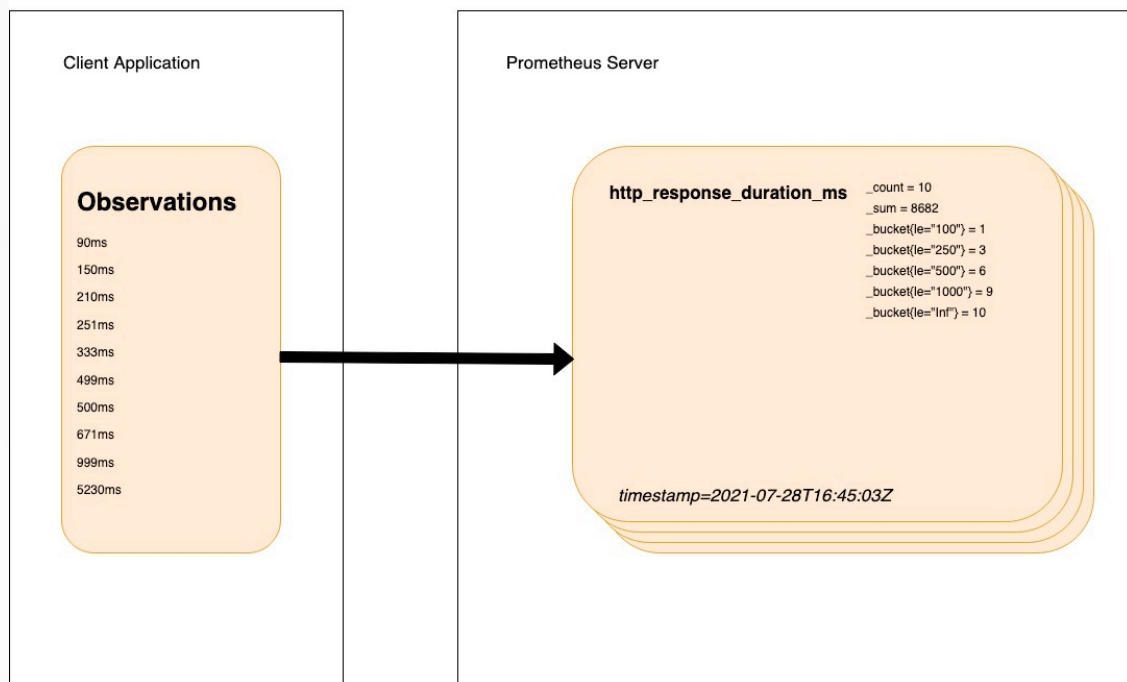
进程数量、内存使用率、温度、并发请求的数量

PromQL 介绍 – 复杂指标类型

Histogram

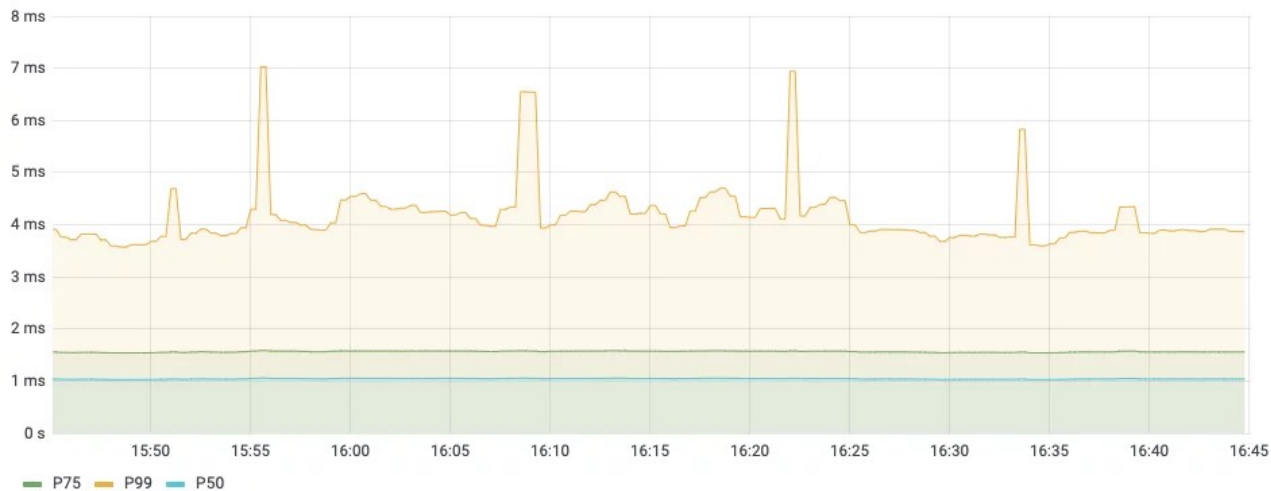
样本的值分布在 bucket 中的数量，命名为
`<basename>_bucket{le="<上边界>"}`

Histograms in Prometheus

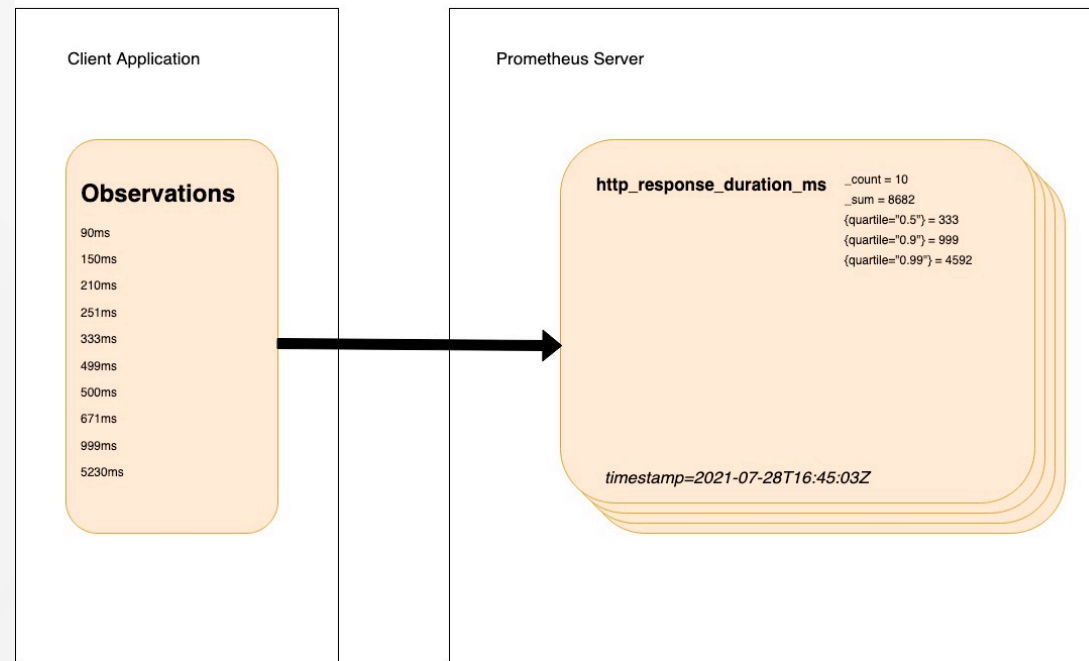


Summary

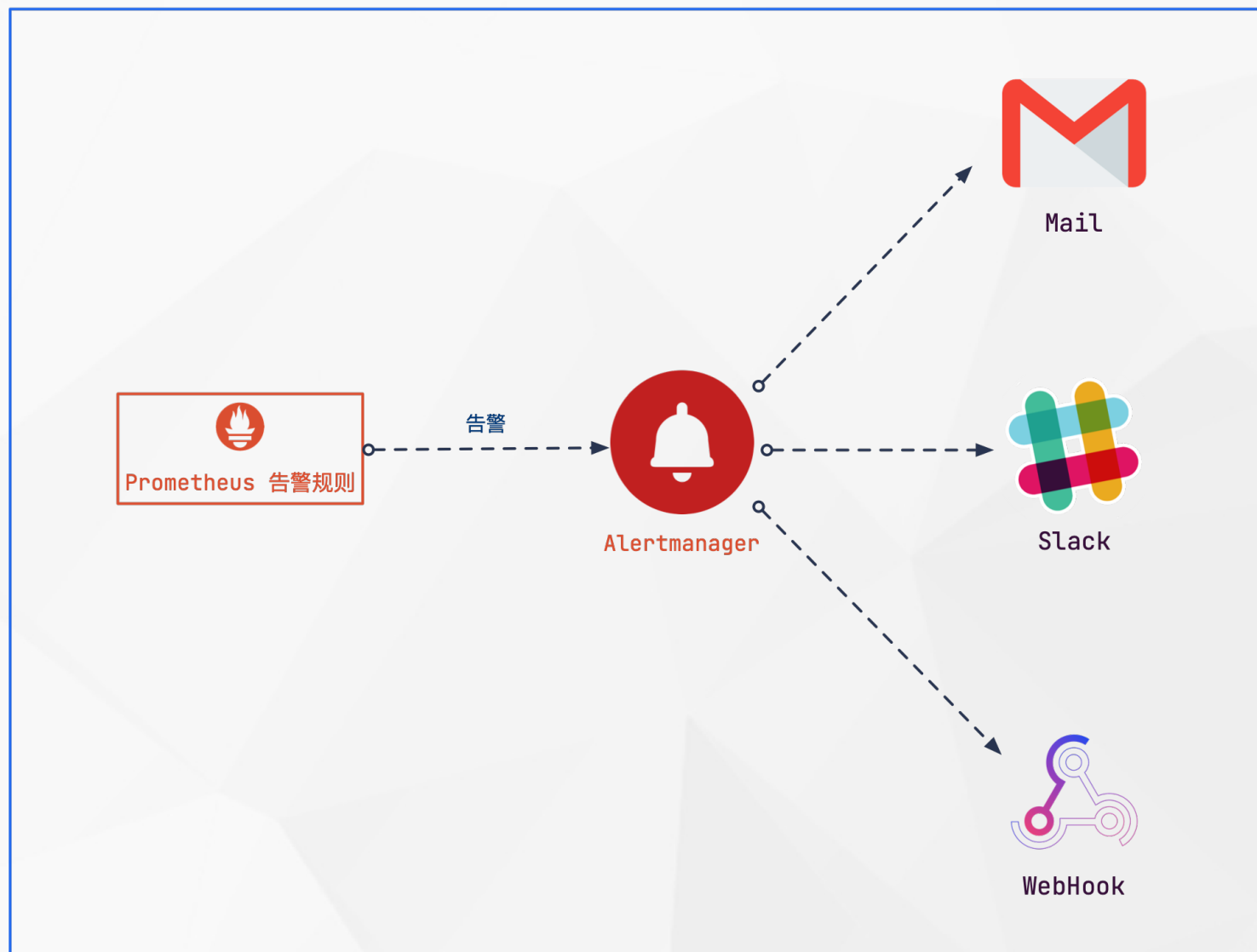
server 端直接算好分位数，不能聚合



Summaries in Prometheus



Prometheus 告警处理 - 架构



Prometheus 告警处理 – Alertmanager 配置

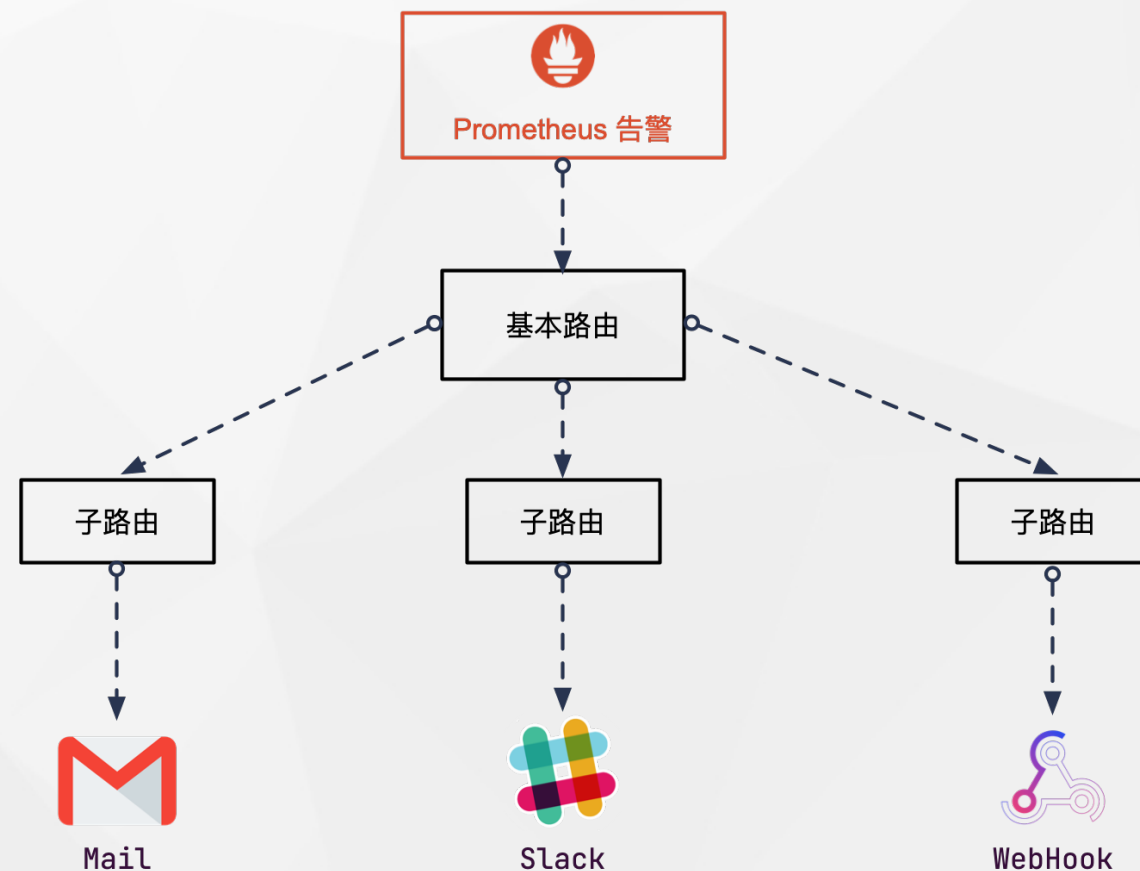
```
# alertmanager.yml

global:
  smtp_smarthost: 'localhost:25'
  smtp_from: 'alertmanager@example.com'
  smtp_require_tls: false

templates:
- '/etc/alertmanager/template/*.tmpl'

route:
  receiver: 'email'
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 4h
  routes:
  - receiver: 'slack'
    group_wait: 10s
    matchers:
    - service=~"mysql|cassandra"
  - receiver: 'webhook'
    group_by: [product, environment]
    matchers:
    - team="frontend"

receivers:
- name: 'email'
  email_configs:
  - to: 'alerts@example.com'
- name: 'slack'
  slack_configs:
  - channel: 'xxxx'
- name: 'webhook'
  webhook_configs:
  - url: 'xxxx'
```



Prometheus 告警处理 – 运行 Alertmanager

```
$ export VERSION=0.23.0
$ wget https://github.com/prometheus/alertmanager/releases/download/v$VERSION/alertmanager-$VERSION.linux-arm64.tar.gz
$ tar zxvf alertmanager-$VERSION.linux-arm64.tar.gz
$ cd alertmanager-$VERSION.linux-arm64

$ ./alertmanager --config.file alertmanager.yml
level=info ts=2021-12-08T14:20:35.457Z caller=main.go:225
msg="Starting Alertmanager" version="(version=0.23.0, branch=HEAD, revision=61046b17771a57cfd4c4a51be370ab930a4d7d54)"
level=info ts=2021-12-08T14:20:35.458Z caller=main.go:226
build_context="(go=go1.16.7, user=root@e21a959be8d2, date=20210825-10:50:35)"
level=info ts=2021-12-08T14:20:35.462Z caller=cluster.go:184
component=cluster msg="setting advertise address explicitly"
addr=192.168.5.15 port=9094
level=info ts=2021-12-08T14:20:35.463Z caller=cluster.go:671
component=cluster msg="Waiting for gossip to settle..." interval=2s
level=info ts=2021-12-08T14:20:35.486Z caller=coordinator.go:113
component=configuration msg="Loading configuration file"
file=alertmanager.yml
level=info ts=2021-12-08T14:20:35.486Z caller=coordinator.go:126
component=configuration msg="Completed loading of configuration file"
file=alertmanager.yml
level=info ts=2021-12-08T14:20:35.488Z caller=main.go:518
msg=Listening address=:9093
level=info ts=2021-12-08T14:20:35.488Z caller=tlscfg.go:191
msg="TLS is disabled." http2=false
```

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s

rule_files:
scrape_configs:
- job_name: "prometheus"
  static_configs:
    - targets: ["localhost:9090"]
# 采集node exporter监控数据
- job_name: "node"
  static_configs:
    - targets: ["localhost:9100"]

alerting:
  alertmanagers:
    - static_configs:
      - targets: ['localhost:9093']
```

Prometheus 告警处理 – 配置告警规则

```
# prometheus.yml

global:
  scrape_interval: 15s
  evaluation_interval: 15s

rule_files:
  - "/etc/prometheus/rules/*_rules.yml"
  - "/etc/prometheus/rules/*_alerts.yml"

scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
    # 采集node exporter监控数据
  - job_name: "node"
    static_configs:
      - targets: ["localhost:9100"]
alerting:
  alertmanagers:
    - static_configs:
        - targets: ['localhost:9093']
```

```
# /etc/prometheus/rules/node_alerts.yml

groups:
- name: node_alerts
  rules:
  - alert: HostOutOfMemory
    expr: node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes
    * 100 > 10
    for: 2m
    labels:
      severity: warning
    annotations:
      summary: Host out of memory (instance {{ $labels.instance }})
      description: "Node memory is filling up (> 10% left)\n VALUE =
{{ $value }}\n LABELS = {{ $labels }}"
```

carbon
carbon.now.sh

- 节点的内存可用率不断变化，每隔一段时间由 `scrape_interval` 定义的时间被 Prometheus 抓取一次，默认是 15 秒。
- 根据每个 `evaluation_interval` 的指标来评估告警规则，默认是 15 秒。
- 当告警表达式为 `true` 时，会创建一个告警并转换到 `Pending` 状态，执行 `for` 语句。
- 在下一个评估周期中，如果告警表达式仍然为 `true`，则检查 `for` 的持续时间。如果超过了持续时间，则告警将转换为 `Firing`，生成通知并将其推送到 `Alertmanager`。
- 如果告警表达式不再为 `true`，则 Prometheus 会将告警规则的状态从 `Pending` 更改为 `Inactive`。

Prometheus 告警处理 – 配置告警规则

Prometheus Alerts Graph Status ▾ Help Classic UI

Inactive (0)

Pending (0)

Firing (1)

Show annotations

/etc/prometheus/rules/node_alerts.yml > node_alerts

firing (1)

▼ HostOutOfMemory (1 active)

name: HostOutOfMemory

expr: node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes * 100 > 10

for: 2m

Labels:

severity: warning

annotations:

description: Node memory is filling up (> 10% left)

VALUE = {{ \$value }}

LABELS = {{ \$labels }}

summary: Host out of memory (instance {{ \$labels.instance }}

Labels	State	Active Since	Value
alertname=HostOutOfMemory instance=localhost:9100 job=node severity=warning	FIRING	2021-12-10T04:58:44.612983603Z	88.61909714001432

Alertmanager Alerts Silences Status Help

New Silence

Filter

Group

Receiver: All ☐ Silenced ☐ Inhibited

alertname="HostOutOfMemory" ×

+

Silence

Custom matcher, e.g. env="production"

+ Expand all groups

— alertname="HostOutOfMemory" +

1 alert

2021-12-10T04:16:29.612Z

+ Info

Source

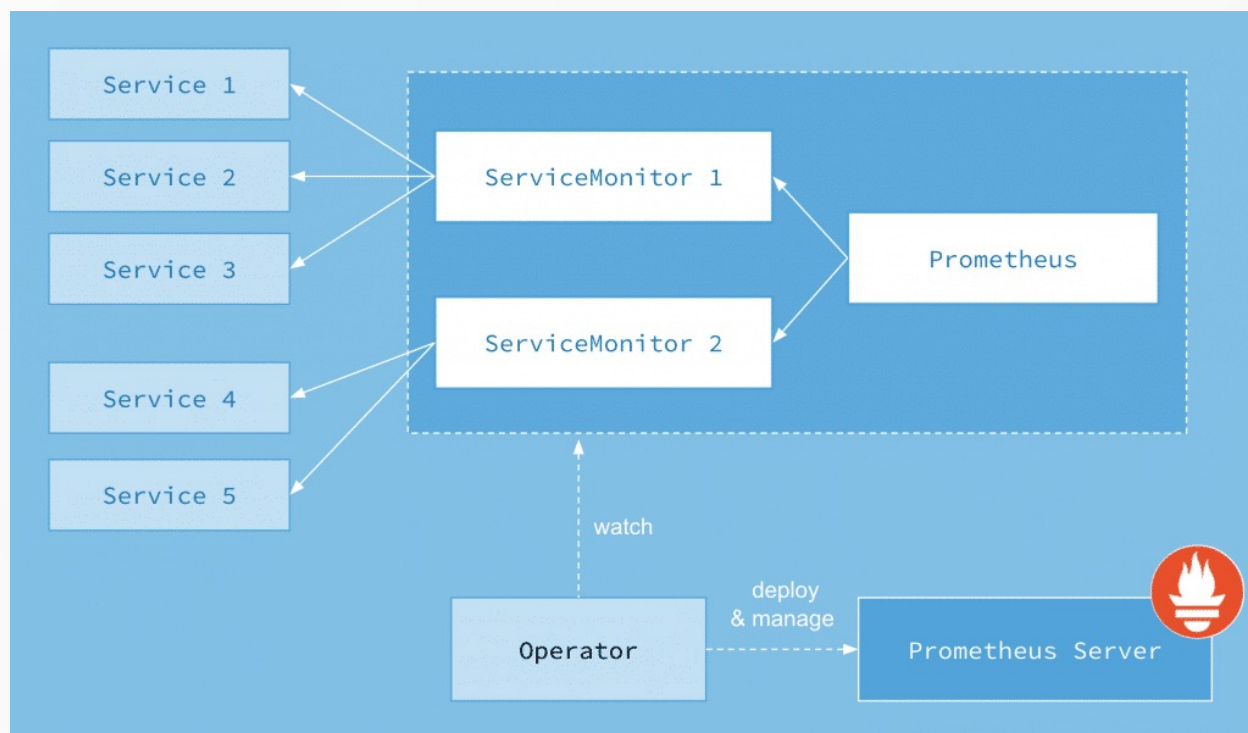
Silence

instance="localhost:9100" +

job="node" +

severity="warning" +

Prometheus Operator – 架构



- **Prometheus** : 声明 Prometheus deployment 期望的状态。
- **Prometheus Server** : Operator 根据自定义资源 Prometheus 类型中定义的内容而部署的 Prometheus Server 集群。
- **Alertmanager** : 声明了 Alertmanager Deployment。
- **ServiceMonitor** : 声明式指定应如何监控 Kubernetes Service , 自动生成相关 Prometheus 抓取配置。
- **Operator** : 根据自定义资源 (CRD) 来部署和管理 Prometheus Server , 同时监控这些自定义资源事件的变化来做相应的处理 , 是整个系统的控制中心。

Prometheus Operator – kube-prometheus

Prometheus-operator

<https://github.com/prometheus-operator/prometheus-operator>

Kube-prometheus

<https://github.com/prometheus-operator/kube-prometheus>

- **Prometheus Operator** : 创建 CRD 自定义的资源对象
- **Highly available Prometheus** : 创建高可用的 Prometheus
- **Highly available Alertmanager** : 创建高可用的告警组件
- **Prometheus node-exporter** : 创建主机的监控组件
- **Prometheus Adapter for Kubernetes Metrics APIs** : 创建自定义监控的指标工具（例如可以通过 Nginx 的 request 来进行应用的自动伸缩）
- **kube-state-metrics** : 监控 K8s 相关资源对象的状态指标
- **Grafana** : 监控面板

Prometheus Operator – kube-prometheus

kube-prometheus stack	Kubernetes 1.18	Kubernetes 1.19	Kubernetes 1.20	Kubernetes 1.21	Kubernetes 1.22
release-0.6	×	✓	×	×	×
release-0.7	×	✓	✓	×	×
release-0.8	×	×	✓	✓	×
release-0.9	×	×	×	✓	✓
main	×	×	×	✓	✓

Prometheus Operator – 安装



KUBESPHERE®
_ by QingCloud



```
$ git clone https://github.com/prometheus-operator/kube-  
prometheus.git  
$ cd kube-prometheus/manifests  
$ mkdir -p serviceMonitor prometheus adapter node-exporter kube-  
state-metrics grafana alertmanager operator blackbox rules  
  
$ mv *Rule.yaml rules  
$ mv *serviceMonitor* serviceMonitor  
$ mv alertmanager-* alertmanager  
$ mv blackboxExporter-* blackbox/  
$ mv grafana-* grafana/  
$ mv kubeStateMetrics-* kube-state-metrics  
$ mv nodeExporter-* node-exporter  
$ mv prometheus-* prometheus/  
$ mv prometheusAdapter-* adapter  
$ mv prometheusOperator-* operator
```

carbon
carbon.now.sh

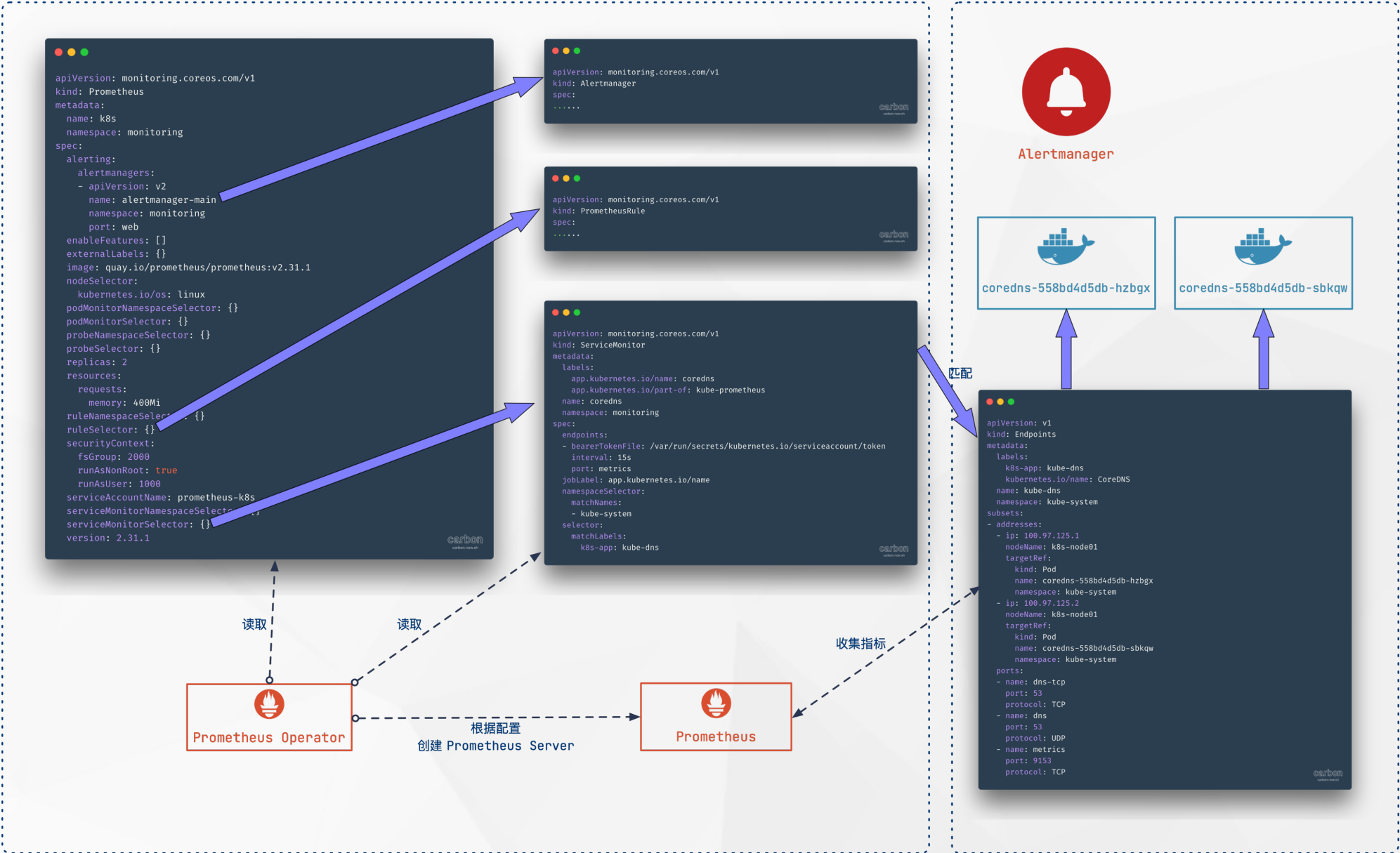
```
$ kubectl apply --server-side -f setup  
$ kubectl apply -f operator  
$ kubectl apply -f adapter/ -f alertmanager/ -f blackbox/ -f grafana/ -f  
kube-state-metrics/ -f node-exporter/ -f prometheus/ -f rules/ -f  
serviceMonitor/
```

carbon
carbon.now.sh

```
$ tree .  
.  
├── adapter  
│   ├── prometheusAdapter-apiService.yaml  
│   ├── prometheusAdapter-clusterRole.yaml  
│   ├── prometheusAdapter-clusterRoleAggregatedMetricsReader.yaml  
│   ├── prometheusAdapter-clusterRoleBinding.yaml  
│   ├── prometheusAdapter-clusterRoleBindingDelegator.yaml  
│   ├── prometheusAdapter-clusterRoleServerResources.yaml  
│   ├── prometheusAdapter-configMap.yaml  
│   ├── prometheusAdapter-deployment.yaml  
│   ├── prometheusAdapter-podDisruptionBudget.yaml  
│   ├── prometheusAdapter-roleBindingAuthReader.yaml  
│   ├── prometheusAdapter-service.yaml  
│   └── prometheusAdapter-serviceAccount.yaml  
├── alertmanager  
│   ├── alertmanager-alertmanager.yaml  
│   ├── alertmanager-podDisruptionBudget.yaml  
│   ├── alertmanager-secret.yaml  
│   ├── alertmanager-service.yaml  
│   └── alertmanager-serviceAccount.yaml  
├── blackbox  
│   ├── blackboxExporter-clusterRole.yaml  
│   ├── blackboxExporter-clusterRoleBinding.yaml  
│   ├── blackboxExporter-configuration.yaml  
│   ├── blackboxExporter-deployment.yaml  
│   ├── blackboxExporter-service.yaml  
│   └── blackboxExporter-serviceAccount.yaml  
├── grafana  
│   ├── grafana-config.yaml  
│   ├── grafana-dashboardDatasources.yaml  
│   ├── grafana-dashboardDefinitions.yaml  
│   ├── grafana-dashboardSources.yaml  
│   ├── grafana-deployment.yaml  
│   ├── grafana-service.yaml  
│   └── grafana-serviceAccount.yaml  
└── kube-state-metrics  
    ├── kubeStateMetrics-clusterRole.yaml  
    ├── kubeStateMetrics-clusterRoleBinding.yaml  
    ├── kubeStateMetrics-deployment.yaml  
    ├── kubeStateMetrics-service.yaml  
    └── kubeStateMetrics-serviceAccount.yaml
```

carbon
carbon.now.sh

Prometheus Operator – 配置解析



Prometheus Operator – 高级配置

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app.kubernetes.io/name: kube-controller-manager
    app.kubernetes.io/part-of: kube-prometheus
  name: kube-controller-manager
  namespace: monitoring
spec:
  endpoints:
    - bearerTokenFile: /var/run/secrets/kubernetes.io/serviceaccount/token
      interval: 30s
      metricRelabelings:
        .....
      port: https-metrics
      scheme: https
      tlsConfig:
        insecureSkipVerify: true
  jobLabel: app.kubernetes.io/name
  namespaceSelector:
    matchNames:
      - kube-system
  selector:
    matchLabels:
      app.kubernetes.io/name: kube-controller-manager
```

```
apiVersion: v1
kind: Service
metadata:
  namespace: kube-system
  name: kube-controller-manager
  labels:
    app.kubernetes.io/name: kube-controller-manager
spec:
  type: ClusterIP
  clusterIP: None
  ports:
    - name: https-metrics
      port: 10257
      targetPort: 10257
      protocol: TCP

apiVersion: v1
kind: Endpoints
metadata:
  labels:
    app.kubernetes.io/name: kube-controller-manager
  name: kube-controller-manager
  namespace: kube-system
subsets:
  - addresses:
      - ip: 192.168.0.5
    ports:
      - name: https-metrics
        port: 10257
        protocol: TCP
```

Prometheus Alerts Graph Status ▾ Help Classic UI

Targets

All Unhealthy Expand All

- serviceMonitor/monitoring/alertmanager-main/0 (1/1 up) [show more](#)
- serviceMonitor/monitoring/alertmanager-main/1 (1/1 up) [show more](#)
- serviceMonitor/monitoring/blackbox-exporter/0 (1/1 up) [show more](#)
- serviceMonitor/monitoring/coredns/0 (2/2 up) [show more](#)
- serviceMonitor/monitoring/grafana/0 (1/1 up) [show more](#)
- serviceMonitor/monitoring/kube-apiserver/0 (1/1 up) [show more](#)
- serviceMonitor/monitoring/kube-controller-manager/0 (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://192.168.0.5:10257/metrics	UP	endpoint="https-metrics" instance="192.168.0.5:10257" job="kube-controller-manager" namespace="kube-system" service="kube-controller-manager"	7.775s ago	16.896ms	



谢谢观看