

StarRocks 存算分离监控部署

说明

StarRocks 存算分离集群在存算一体基础上增加了众多与后端存储相关的监控指标，本文章主要提供这些指标在 Prometheus 和 Grafana 上的部署以及各字段含义说明，存算一体的监控部署可参考 https://docs.starrocks.io/zh-cn/latest/administration/Monitor_and_Alert，这里不再赘述。

部署 Prometheus 和 Grafana

详情内容请参考：https://docs.starrocks.io/zh-cn/latest/administration/Monitor_and_Alert

部署 Prometheus

下载并解压 Prometheus

```
1 # 注意需要使用高版本的 Prometheus
2 curl -LO https://github.com/prometheus/prometheus/releases/download/v2.45.1/prom
3 tar -xf prometheus-2.29.1.linux-amd64.tar.gz
```

准备配置文件：

```
1 # prometheus.yml
2 # my global config
3 global:
4   scrape_interval:     15s # 全局的采集间隔，默认是 1m，这里设置为 15s
5   evaluation_interval: 15s # 全局的规则触发间隔，默认是 1m，这里设置 15s
6
7 scrape_configs:
8   # The job name is added as a label `job=<job_name>` to any timeseries scraped
9   - job_name: 'StarRocks_Cluster01' # 每一个集群称之为一个job，可以自定义名字作为Star
10     metrics_path: '/metrics'      # 指定获取监控项目的Restful Api
11     static_configs:
12       - targets: ['172.26.81.36:8030'] # fe http port
13         labels:
14           group: fe
15       - targets: ['172.26.81.37:8040', '172.26.81.38:8040', '172.26.81.39:8040']
16         labels:
```

启动 Prometheus

```
1 nohup ./prometheus \  
2   --config.file="./prometheus.yml" \  
3   --web.listen-address=":9090" \  
4   --log.level="info" &
```

访问 Prometheus 的 9090 端口，如果显示以下内容则表示服务成功启动。

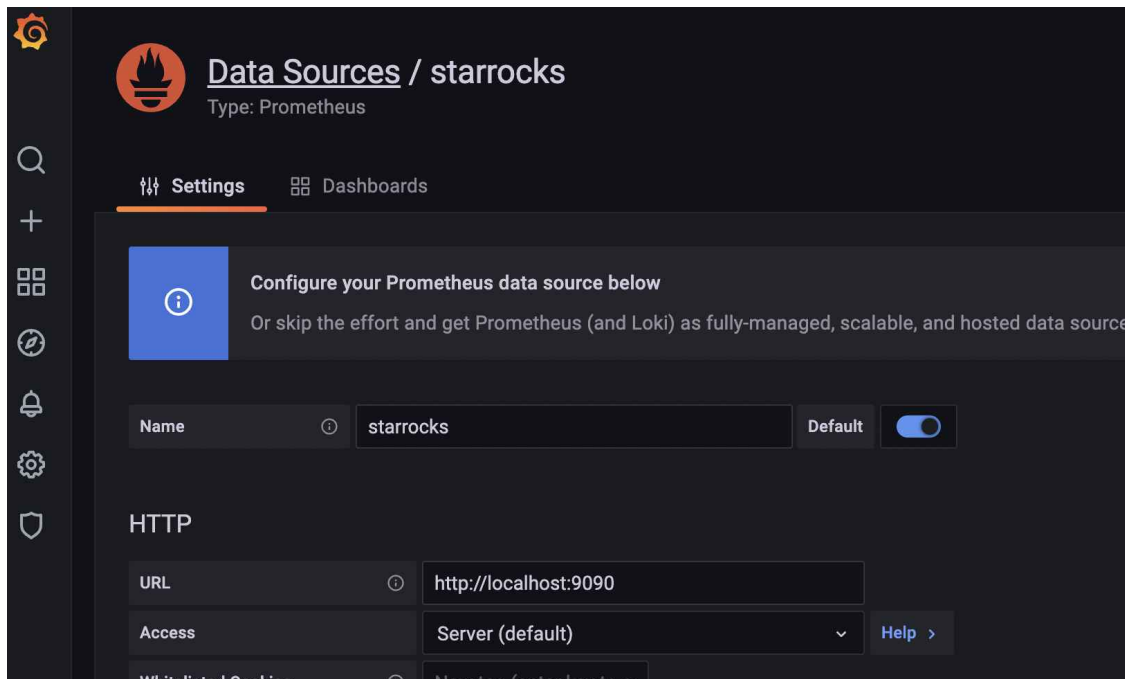
Prometheus Alerts Graph Status ▾ Help Classic UI					
Targets					
All Unhealthy Collapse All					
StarRocks_Cluster01 (4/4 up) show less					
Endpoint	State	Labels	Last Scrape	Scrape Duration	E
http://172.26.81.39:8040/metrics	UP	group="be" instance="172.26.81.39:8040" job="StarRocks_Cluster01"	11.191s ago	7.400ms	
http://172.26.81.36:8030/metrics	UP	group="fe" instance="172.26.81.36:8030" job="StarRocks_Cluster01"	14.546s ago	3.927ms	
http://172.26.81.37:8040/metrics	UP	group="be" instance="172.26.81.37:8040" job="StarRocks_Cluster01"	6.810s ago	7.227ms	
http://172.26.81.38:8040/metrics	UP	group="be" instance="172.26.81.38:8040" job="StarRocks_Cluster01"	8.229s ago	7.573ms	

部署 Grafana

```
1 curl -LO https://dl.grafana.com/oss/release/grafana-10.1.4.linux-amd64.tar.gz  
2 tar -zxf grafana-8.0.6.linux-amd64.tar.gz  
3  
4 nohup ./bin/grafana-server --config="./conf/defaults.ini" &
```

为 Grafana 配置 Prometheus 作为数据源。登录用户名和密码都是 admin。

依次点击 Configuration，Data sources，Add data source，和 Prometheus。在下面的页面配置：Name、URL、Access。



添加 Dashboard

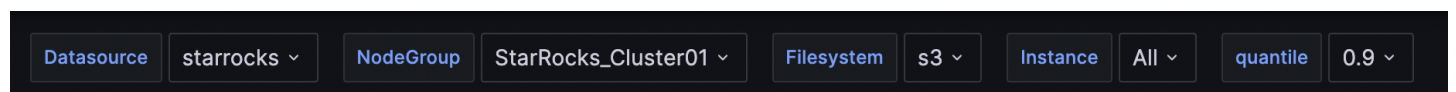
添加 Shared Data Dashboard

导入 JSON 文件，文件内容参考后面附件中的 shared_data.json。

添加 Starlet Dashboard

导入 JSON 文件，文件内容参考后面附件中的 starlet_fslib.json。

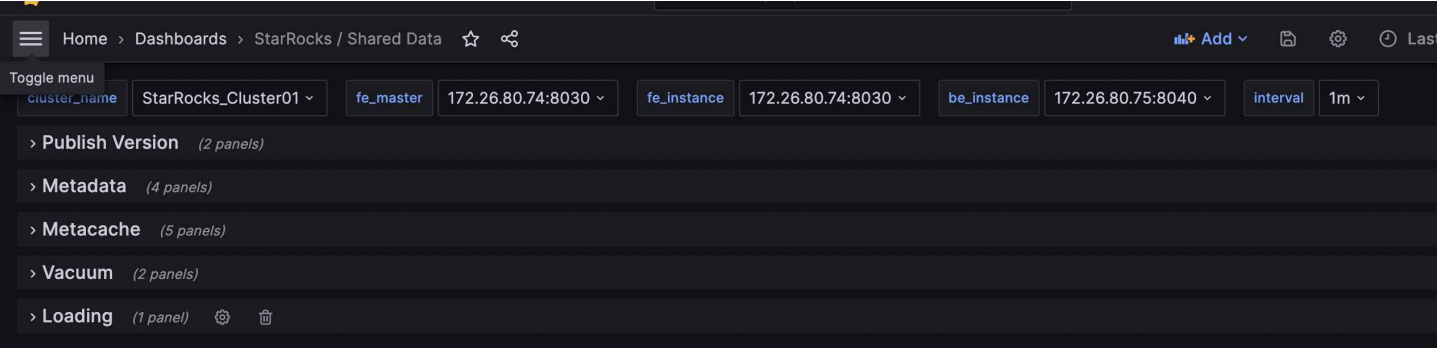
注意，查看 dashboard 时，需要选择正确的 **Datasource** 和 **NodeGroup**。注意：不能选择 All。



理解 Dashboard

Shared Data Dashboard

Shared Data Dashboard 总共包含五个监控类别，如下：



Publish Version

名称	描述
Latency / QPS	Public Version \$quantile 分位延迟，平均延迟，以及 QPS
Queued Tasks	处于排队中的 Public Version 任务数量

Metadata

名称	描述
Get Tablet Metadata	Get Tablet Metadata 的 \$quantile 分位延迟，平均延迟，以及 QPS
Put Tablet Metadata	Put Tablet Metadata 的 \$quantile 分位延迟，平均延迟，以及 QPS
Get Txn Log	Get Txn Log 的 \$quantile 分位延迟，平均延迟，以及 QPS
Put Txn Log	Put Txn Log 的 \$quantile 分位延迟，平均延迟，以及 QPS

Metacache

名称	描述
Metacache Usage	Metacache 的利用率
Delvec Cache Miss Per Minute	Delvec Cache 每分钟 Miss 的次数
Metadata Cache Miss Per Minute	Metadata Cache 每分钟 Miss 的次数
Txn Log Cache Miss Per Minute	Txn Log Cache 每分钟 Miss 的次数
Segment Cache Miss Per Minute	Segment Cache 每分钟 Miss 的次数

Vacuum

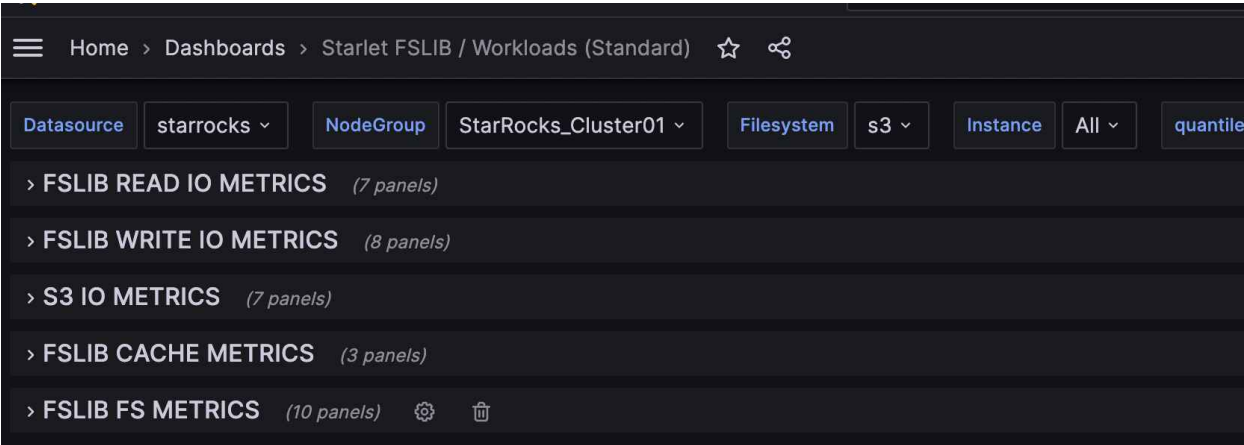
名称	描述
Vacuum Deletes	Vacuum Deletes 的 \$quantile 分位延迟，平均延迟，以及 QPS
Errors	Vacuum Deletes 操作失败的次数

Loading

名称	描述
Queue Size	BE async delta writer 的队列长度

Starlet Dashboard

Starlet Dashboard 总共包含五个监控类别，如下：



在 Grafana 图表中可以按需选择观察的文件系统类型（例如 S3、HDFS、Posix 等）。这里我们以 S3 为例，详细介绍各指标的含义。

FSLIB READ IO METRICS

名称	指标类型	描述
fslib read io_latency (quantile)	Histogram	读取 S3 的 \$quantile 分位延迟
fslib read io_latency (average)	Counter	读取 S3 的平均延迟
fslib total read data	Counter	累计读取 S3 的数据总量
fslib read iosize (quantile)	Histogram	读取 S3 的 \$quantile 分位 IO 大小

fslib read iosize (average)	Counter	读取 S3 的平均 IO 大小
fslib read throughput	Counter	每秒钟读取 S3 的 IO 吞吐
fslib read iops	Counter	每秒读取 S3 的 IO 次数

FSLIB WRITE IO METRICS

名称	指标类型	描述
fslib write io_latency (quantile)	Histogram	应用写入的 \$quantile 分位延迟，注意这里只是写入缓冲区，因此观察到的值会比较低
fslib write io_latency (average)	Counter	应用写入的平均延迟，注意这里只是写入缓冲区，因此观察到的值会比较低
fslib total write data	Counter	累计写入的数据总量
fslib write iosize (quantile)	Histogram	应用每次写入请求的 \$quantile 分位大小
fslib write iosize (average)	Counter	应用每次写入请求的平均大小
fslib write throughput	Counter	应用的写入吞吐

S3 IO METRICS

该栏包含 S3 的各种详细指标。

名称	指标类型	描述
fslib s3 single upload iops	Counter	S3 Put Object 的每秒调用次数
fslib s3 single upload iosize (quantile)	Histogram	S3 Put Object 的 \$quantile 分位大小
fslib s3 single upload latency (quantile)	Histogram	S3 Put Object 的 \$quantile 分位延迟
fslib s3 multi upload iops	Counter	S3 Multi Upload Object 的每秒调用次数
fslib s3 multi upload iosize (quantile)	Histogram	S3 Multi Upload Object 的 \$quantile 分位大小
fslib s3 multi upload latency (quantile)	Histogram	S3 Multi Upload Object 的 \$quantile 分位延迟
fslib s3 complete multi upload latency (quantile)	Histogram	S3 Complete Multi Upload Object 的 \$quantile 分位延迟

FSLIB CACHE METRICS


名称	指标类型	描述
fslib cache hit ratio	Counter	缓存访问的命中率
fslib cache hits/misses	Counter	每秒打开缓存文件的命中次数

FSLIB FS METRICS


名称	指标类型	描述
fslib alive fs instances count	Guage	文件系统实例存活数。
fslib open files	Counter	累计打开的文件数量
fslib create files	Counter	每秒平均创建的文件数
filesystem meta operations	Counter	每秒平均 list 文件目录的次数
fslib async caches	Counter	累计异步缓存的文件数量
fslib create files (TOTAL)	Counter	累计创建的文件数量
fslib async tasks	Counter	累计排队中的异步任务
fslib async writeback		没有该指标
fslib list latency (quantile)		没有该指标
fslib list latency (average)		没有该指标

附件



 shared_data.json



 starlet_fslib.json